

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of

Inventors: Jun HIRANO, et al.

Application No.: 10/585,943

Filed: July 13, 2006

For: DYNAMIC NETWORK MANAGEMENT APPARATUS AND
DYNAMIC NETWORK MANAGEMENT METHOD

INFORMATION DISCLOSURE STATEMENT

Assistant Commissioner of Patents
Washington, DC 20231

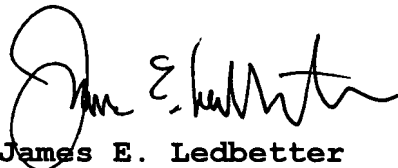
Dear Sir:

Pursuant to Rules 56 and 99, Applicants hereby call the attention of the Patent Office to the documents listed on the attached Form PTO 1449. The Perkins et al. reference is discussed in the present application and cited at page 7, line 4 et seq. of the Specification. The DARPA reference is discussed in the present application and cited at page 7, line 6 et seq. of the Specification. The Johnson et al. reference is discussed in the present application and cited at page 7, line 8 et seq. of the Specification. The Deering et al. reference is discussed in the present application and cited at page 7, line 12 et seq. of the Specification. The W. Simpson reference is discussed in the present application and cited at page 7, line 15 et seq. of the Specification. The Conta et al. reference is discussed in the

present application and cited at page 7, line 17 et seq. of the Specification. The Devarapalli et al. reference is discussed in the present application and cited at page 7, line 20 et seq. of the Specification.

Applicants present this art so that the Patent Office may, in the first instance, determine any relevancy thereof to the presently claimed invention, see Beckman Instruments, Inc. v. Chemtronics, Inc., 439 F.2d 1369, 1380, 165 USPQ 355, 364 (5th Cir. 1970). Also see Patent Office Rules 104 and 106. Applicants respectfully request that this art be expressly considered during the prosecution of this application and made of record herein and appear among the "References Cited" on any patent to issue herefrom.

Respectfully submitted,



James E. Ledbetter
Registration No. 28,732

Date: May 30, 2007

JEL/jpf

ATTORNEY DOCKET NO. L8638.06112
STEVENS, DAVIS, MILLER & MOSHER, L.L.P.
1615 L STREET, NW, Suite 850
WASHINGTON, DC 20043-4387
Telephone: (202) 785-0100
Facsimile: (202) 408-5200

FORM PTO-1449 U.S. Department of Commerce
(Rev. 4/92) Patent and Trademark Office

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use several sheets if necessary)

ATTY. DOCKET NO.

L8638.06112

SERIAL NO.

10/585,943

APPLICANT

Jun HIRANO, et al.

FILING DATE

July 13, 2006

GROUP

Unassigned

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUBCLASS	TRANSLATION	
					YES	NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

C. Perkins, et al.: "IP Mobility Support for IPv4," IETF RCF 3344, August 2002, pp. 1-99.

DARPA, "Internet Protocol," IETF RFC 791, September 1981, pp. i-iii and 1-45.

D. Johnson, et al.: "Mobility Support in IPv6," IETF Internet Draft, draft-ietf-mobileip-ipv6-24.txt, Work in Progress, June 2003, pp. 1-172.

S. Deering, et al.: "Internet Protocol, Version 6 (IPv6) Specification," IETF RFC 2460, December 1998, pp. 1-39.

W. Simpson: "IP in IP Tunneling," IETF RFC 1853, October 1995, pp. 1-8.

A. Conta, et al.: "Generic Packet Tunneling in IPv6 Specification," IETF RFC 2473, December 1998, pp. 1-36.

V. Devarapalli, et al.: "Nemo Basic Support Protocol," IETF Internet Draft, draft-ietf-nemo-basic-support-01.txt, September 2003, pp. 1-33.

EXAMINER: Initial if citation is considered, draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

IP Mobility Support for IPv4

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

This document specifies protocol enhancements that allow transparent routing of IP datagrams to mobile nodes in the Internet. Each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a care-of address, which provides information about its current point of attachment to the Internet. The protocol provides for registering the care-of address with a home agent. The home agent sends datagrams destined for the mobile node through a tunnel to the care-of address. After arriving at the end of the tunnel, each datagram is then delivered to the mobile node.

Contents

1. Introduction	3
1.1. Protocol Requirements	4
1.2. Goals	4
1.3. Assumptions	5
1.4. Applicability	5
1.5. New Architectural Entities	5
1.6. Terminology	6
1.7. Protocol Overview	9
1.8. Message Format and Protocol Extensibility	13
1.9. Type-Length-Value Extension Format for Mobile IP Extensions	15
1.10. Long Extension Format	16

1.11. Short Extension Format	16
2. Agent Discovery	17
2.1. Agent Advertisement	18
2.1.1. Mobility Agent Advertisement Extension	20
2.1.2. Prefix-Lengths Extension	22
2.1.3. One-byte Padding Extension	22
2.2. Agent Solicitation	23
2.3. Foreign Agent and Home Agent Considerations	23
2.3.1. Advertised Router Addresses	24
2.3.2. Sequence Numbers and Rollover Handling	24
2.4. Mobile Node Considerations	25
2.4.1. Registration Required	26
2.4.2. Move Detection	26
2.4.3. Returning Home	27
2.4.4. Sequence Numbers and Rollover Handling	28
3. Registration	28
3.1. Registration Overview	29
3.2. Authentication	30
3.3. Registration Request	30
3.4. Registration Reply	33
3.5. Registration Extensions	36
3.5.1. Computing Authentication Extension Values	36
3.5.2. Mobile-Home Authentication Extension	37
3.5.3. Mobile-Foreign Authentication Extension	37
3.5.4. Foreign-Home Authentication Extension	38
3.6. Mobile Node Considerations	38
3.6.1. Sending Registration Requests	40
3.6.2. Receiving Registration Replies	44
3.6.3. Registration Retransmission	47
3.7. Foreign Agent Considerations	47
3.7.1. Configuration and Registration Tables	48
3.7.2. Receiving Registration Requests	49
3.7.3. Receiving Registration Replies	52
3.8. Home Agent Considerations	54
3.8.1. Configuration and Registration Tables	55
3.8.2. Receiving Registration Requests	56
3.8.3. Sending Registration Replies	59
4. Routing Considerations	62
4.1. Encapsulation Types	62
4.2. Unicast Datagram Routing	62
4.2.1. Mobile Node Considerations	62
4.2.2. Foreign Agent Considerations	63
4.2.3. Home Agent Considerations	64
4.3. Broadcast Datagrams	66
4.4. Multicast Datagram Routing	66
4.5. Mobile Routers	67
4.6. ARP, Proxy ARP, and Gratuitous ARP	69
5. Security Considerations	73

5.1. Message Authentication Codes	73
5.2. Areas of Security Concern in this Protocol	73
5.3. Key Management	74
5.4. Picking Good Random Numbers	74
5.5. Privacy	74
5.6. Ingress Filtering	75
5.7. Replay Protection for Registration Requests	75
5.7.1. Replay Protection using Timestamps	75
5.7.2. Replay Protection using Nonces	77
6. IANA Considerations	77
6.1. Mobile IP Message Types	78
6.2. Extensions to RFC 1256 Router Advertisement	78
6.3. Extensions to Mobile IP Registration Messages	79
6.4. Code Values for Mobile IP Registration Reply Messages.	79
7. Acknowledgments	80
A. Patent Issues	82
B. Link-Layer Considerations	82
C. TCP Considerations	83
C.1. TCP Timers	83
C.2. TCP Congestion Management	83
D. Example Scenarios	84
D.1. Registering with a Foreign Agent Care-of Address	84
D.2. Registering with a Co-Located Care-of Address	84
D.3. Deregistration	85
E. Applicability of Prefix-Lengths Extension	86
F. Interoperability Considerations	86
G. Changes since RFC 2002	87
G.1. Major Changes	87
G.2. Minor Changes	89
G.3. Changes since revision 04 of RFC2002bis	91
H. Example Messages	92
H.1. Example ICMP Agent Advertisement Message Format	92
H.2. Example Registration Request Message Format	93
H.3. Example Registration Reply Message Format	94
References	94
Authors' Addresses	98
Full Copyright Statement	99

1. Introduction

IP version 4 assumes that a node's IP address uniquely identifies the node's point of attachment to the Internet. Therefore, a node must be located on the network indicated by its IP address in order to receive datagrams destined to it; otherwise, datagrams destined to the node would be undeliverable. For a node to change its point of attachment without losing its ability to communicate, currently one of the two following mechanisms must typically be employed:

- a) the node must change its IP address whenever it changes its point of attachment, or
- b) host-specific routes must be propagated throughout much of the Internet routing fabric.

Both of these alternatives are often unacceptable. The first makes it impossible for a node to maintain transport and higher-layer connections when the node changes location. The second has obvious and severe scaling problems, especially relevant considering the explosive growth in sales of notebook (mobile) computers.

A new, scalable, mechanism is required for accommodating node mobility within the Internet. This document defines such a mechanism, which enables nodes to change their point of attachment to the Internet without changing their IP address.

Changes between this revised specification for Mobile IP and the original specifications (see [33, 32, 34, 43, 8]) are detailed in the appendix section G.

1.1. Protocol Requirements

A mobile node must be able to communicate with other nodes after changing its link-layer point of attachment to the Internet, yet without changing its IP address.

A mobile node must be able to communicate with other nodes that do not implement these mobility functions. No protocol enhancements are required in hosts or routers that are not acting as any of the new architectural entities introduced in Section 1.5.

All messages used to update another node as to the location of a mobile node must be authenticated in order to protect against remote redirection attacks.

1.2. Goals

The link by which a mobile node is directly attached to the Internet may often be a wireless link. This link may thus have a substantially lower bandwidth and higher error rate than traditional wired networks. Moreover, mobile nodes are likely to be battery powered, and minimizing power consumption is important. Therefore, the number of administrative messages sent over the link by which a mobile node is directly attached to the Internet should be minimized, and the size of these messages should be kept as small as is reasonably possible.

1.3. Assumptions

The protocols defined in this document place no additional constraints on the assignment of IP addresses. That is, a mobile node can be assigned an IP address by the organization that owns the machine.

This protocol assumes that mobile nodes will generally not change their point of attachment to the Internet more frequently than once per second.

This protocol assumes that IP unicast datagrams are routed based on the destination address in the datagram header (and not, for example, by source address).

1.4. Applicability

Mobile IP is intended to enable nodes to move from one IP subnet to another. It is just as suitable for mobility across homogeneous media as it is for mobility across heterogeneous media. That is, Mobile IP facilitates node movement from one Ethernet segment to another as well as it accommodates node movement from an Ethernet segment to a wireless LAN, as long as the mobile node's IP address remains the same after such a movement.

One can think of Mobile IP as solving the "macro" mobility management problem. It is less well suited for more "micro" mobility management applications — for example, handoff amongst wireless transceivers, each of which covers only a very small geographic area. As long as node movement does not occur between points of attachment on different IP subnets, link-layer mechanisms for mobility (i.e., link-layer handoff) may offer faster convergence and far less overhead than Mobile IP.

1.5. New Architectural Entities

Mobile IP introduces the following new functional entities:

Mobile Node

A host or router that changes its point of attachment from one network or subnetwork to another. A mobile node may change its location without changing its IP address; it may continue to communicate with other Internet nodes at any location using its (constant) IP address, assuming link-layer connectivity to a point of attachment is available.

Home Agent

A router on a mobile node's home network which tunnels datagrams for delivery to the mobile node when it is away from home, and maintains current location information for the mobile node.

Foreign Agent

A router on a mobile node's visited network which provides routing services to the mobile node while registered. The foreign agent detunnels and delivers datagrams to the mobile node that were tunneled by the mobile node's home agent. For datagrams sent by a mobile node, the foreign agent may serve as a default router for registered mobile nodes.

A mobile node is given a long-term IP address on a home network. This home address is administered in the same way as a "permanent" IP address is provided to a stationary host. When away from its home network, a "care-of address" is associated with the mobile node and reflects the mobile node's current point of attachment. The mobile node uses its home address as the source address of all IP datagrams that it sends, except where otherwise described in this document for datagrams sent for certain mobility management functions (e.g., as in Section 3.6.1.1).

1.6. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [4].

In addition, this document frequently uses the following terms:

Authorization-enabling extension

An authentication which makes a (registration) message acceptable to the ultimate recipient of the registration message. An authorization-enabling extension **MUST** contain an SPI.

In this document, all uses of authorization-enabling extension refer to authentication extensions that enable the Registration Request message to be acceptable to the home agent. Using additional protocol structures specified outside of this document, it may be possible for the mobile node to provide authentication of its registration to the

home agent, by way of another authenticating entity within the network that is acceptable to the home agent (for example, see RFC 2794 [6]).

Agent Advertisement

An advertisement message constructed by attaching a special Extension to a router advertisement [10] message.

Authentication

The process of verifying (using cryptographic techniques, for all applications in this specification) the identity of the originator of a message.

Care-of Address

The termination point of a tunnel toward a mobile node, for datagrams forwarded to the mobile node while it is away from home. The protocol can use two different types of care-of address: a "foreign agent care-of address" is an address of a foreign agent with which the mobile node is registered, and a "co-located care-of address" is an externally obtained local address which the mobile node has associated with one of its own network interfaces.

Correspondent Node

A peer with which a mobile node is communicating. A correspondent node may be either mobile or stationary.

Foreign Network

Any network other than the mobile node's Home Network.

Gratuitous ARP

An ARP packet sent by a node in order to spontaneously cause other nodes to update an entry in their ARP cache [45]. See section 4.6.

Home Address

An IP address that is assigned for an extended period of time to a mobile node. It remains unchanged regardless of where the node is attached to the Internet.

Home Network

A network, possibly virtual, having a network prefix matching that of a mobile node's home address. Note that standard IP routing mechanisms will deliver datagrams destined to a mobile node's Home Address to the mobile node's Home Network.

Link

A facility or medium over which nodes can communicate at the link layer. A link underlies the network layer.

Link-Layer Address

The address used to identify an endpoint of some communication over a physical link. Typically, the Link-Layer address is an interface's Media Access Control (MAC) address.

Mobility Agent

Either a home agent or a foreign agent.

Mobility Binding

The association of a home address with a care-of address, along with the remaining lifetime of that association.

Mobility Security Association

A collection of security contexts, between a pair of nodes, which may be applied to Mobile IP protocol messages exchanged between them. Each context indicates an authentication algorithm and mode (Section 5.1), a secret (a shared key, or appropriate public/private key pair), and a style of replay protection in use (Section 5.7).

Node

A host or a router.

Nonce

A randomly chosen value, different from previous choices, inserted in a message to protect against replays.

Security Parameter Index (SPI)

An index identifying a security context between a pair of nodes among the contexts available in the Mobility Security Association. SPI values 0 through 255 are reserved and MUST NOT be used in any Mobility Security Association.

Tunnel

The path followed by a datagram while it is encapsulated. The model is that, while it is encapsulated, a datagram is routed to a knowledgeable decapsulating agent, which decapsulates the datagram and then correctly delivers it to its ultimate destination.

Virtual Network

A network with no physical instantiation beyond a router (with a physical network interface on another network). The router (e.g., a home agent) generally advertises reachability to the virtual network using conventional routing protocols.

Visited Network

A network other than a mobile node's Home Network, to which the mobile node is currently connected.

Visitor List

The list of mobile nodes visiting a foreign agent.

1.7. Protocol Overview

The following support services are defined for Mobile IP:

Agent Discovery

Home agents and foreign agents may advertise their availability on each link for which they provide service. A newly arrived mobile node can send a solicitation on the link to learn if any prospective agents are present.

Registration

When the mobile node is away from home, it registers its care-of address with its home agent. Depending on its method of attachment, the mobile node will register either

directly with its home agent, or through a foreign agent which forwards the registration to the home agent.

silently discard

The implementation discards the datagram without further processing, and without indicating an error to the sender. The implementation **SHOULD** provide the capability of logging the error, including the contents of the discarded datagram, and **SHOULD** record the event in a statistics counter.

The following steps provide a rough outline of operation of the Mobile IP protocol:

- Mobility agents (i.e., foreign agents and home agents) advertise their presence via Agent Advertisement messages (Section 2). A mobile node may optionally solicit an Agent Advertisement message from any locally attached mobility agents through an Agent Solicitation message.
- A mobile node receives these Agent Advertisements and determines whether it is on its home network or a foreign network.
- When the mobile node detects that it is located on its home network, it operates without mobility services. If returning to its home network from being registered elsewhere, the mobile node deregisters with its home agent, through exchange of a Registration Request and Registration Reply message with it.
- When a mobile node detects that it has moved to a foreign network, it obtains a care-of address on the foreign network. The care-of address can either be determined from a foreign agent's advertisements (a foreign agent care-of address), or by some external assignment mechanism such as DHCP [13] (a co-located care-of address).
- The mobile node operating away from home then registers its new care-of address with its home agent through exchange of a Registration Request and Registration Reply message with it, possibly via a foreign agent (Section 3).
- Datagrams sent to the mobile node's home address are intercepted by its home agent, tunneled by the home agent to the mobile node's care-of address, received at the tunnel endpoint (either at a foreign agent or at the mobile node itself), and finally delivered to the mobile node (Section 4.2.3).

- In the reverse direction, datagrams sent by the mobile node are generally delivered to their destination using standard IP routing mechanisms, not necessarily passing through the home agent.

When away from home, Mobile IP uses protocol tunneling to hide a mobile node's home address from intervening routers between its home network and its current location. The tunnel terminates at the mobile node's care-of address. The care-of address must be an address to which datagrams can be delivered via conventional IP routing. At the care-of address, the original datagram is removed from the tunnel and delivered to the mobile node.

Mobile IP provides two alternative modes for the acquisition of a care-of address:

- a) A "foreign agent care-of address" is a care-of address provided by a foreign agent through its Agent Advertisement messages. In this case, the care-of address is an IP address of the foreign agent. In this mode, the foreign agent is the endpoint of the tunnel and, upon receiving tunneled datagrams, decapsulates them and delivers the inner datagram to the mobile node. This mode of acquisition is preferred because it allows many mobile nodes to share the same care-of address and therefore does not place unnecessary demands on the already limited IPv4 address space.
- b) A "co-located care-of address" is a care-of address acquired by the mobile node as a local IP address through some external means, which the mobile node then associates with one of its own network interfaces. The address may be dynamically acquired as a temporary address by the mobile node such as through DHCP [13], or may be owned by the mobile node as a long-term address for its use only while visiting some foreign network. Specific external methods of acquiring a local IP address for use as a co-located care-of address are beyond the scope of this document. When using a co-located care-of address, the mobile node serves as the endpoint of the tunnel and itself performs decapsulation of the datagrams tunneled to it.

The mode of using a co-located care-of address has the advantage that it allows a mobile node to function without a foreign agent, for example, in networks that have not yet deployed a foreign agent. It does, however, place additional burden on the IPv4 address space because it requires a pool of addresses within the foreign network to

be made available to visiting mobile nodes. It is difficult to efficiently maintain pools of addresses for each subnet that may permit mobile nodes to visit.

It is important to understand the distinction between the care-of address and the foreign agent functions. The care-of address is simply the endpoint of the tunnel. It might indeed be an address of a foreign agent (a foreign agent care-of address), but it might instead be an address temporarily acquired by the mobile node (a co-located care-of address). A foreign agent, on the other hand, is a mobility agent that provides services to mobile nodes. See Sections 3.7 and 4.2.2 for additional details.

For example, figure 1 illustrates the routing of datagrams to and from a mobile node away from home, once the mobile node has registered with its home agent. In figure 1, the mobile node is using a foreign agent care-of address, not a co-located care-of address.

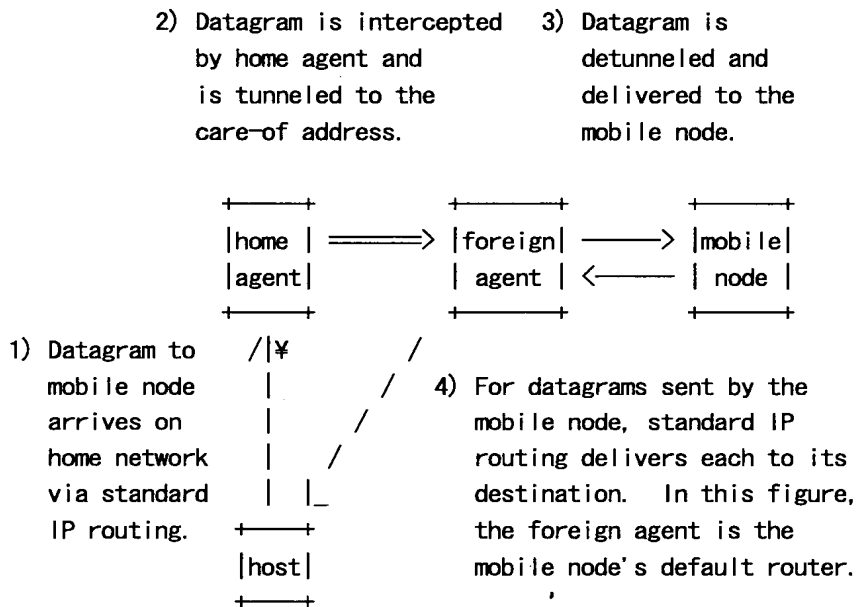


Figure 1: Operation of Mobile IPv4

A home agent **MUST** be able to attract and intercept datagrams that are destined to the home address of any of its registered mobile nodes. Using the proxy and gratuitous ARP mechanisms described in Section 4.6, this requirement can be satisfied if the home agent has a network interface on the link indicated by the mobile node's home address. Other placements of the home agent relative to the mobile node's home location **MAY** also be possible using other mechanisms for intercepting datagrams destined to the mobile node's home address. Such placements are beyond the scope of this document.

Similarly, a mobile node and a prospective or current foreign agent **MUST** be able to exchange datagrams without relying on standard IP routing mechanisms; that is, those mechanisms which make forwarding decisions based upon the network-prefix of the destination address in the IP header. This requirement can be satisfied if the foreign agent and the visiting mobile node have an interface on the same link. In this case, the mobile node and foreign agent simply bypass their normal IP routing mechanism when sending datagrams to each other, addressing the underlying link-layer packets to their respective link-layer addresses. Other placements of the foreign agent relative to the mobile node **MAY** also be possible using other mechanisms to exchange datagrams between these nodes, but such placements are beyond the scope of this document.

If a mobile node is using a co-located care-of address (as described in (b) above), the mobile node **MUST** be located on the link identified by the network prefix of this care-of address. Otherwise, datagrams destined to the care-of address would be undeliverable.

1.8. Message Format and Protocol Extensibility

Mobile IP defines a set of new control messages, sent with UDP [37] using well-known port number 434. The following two message types are defined in this document:

- 1 Registration Request
- 3 Registration Reply

Up-to-date values for the message types for Mobile IP control messages are specified in the most recent "Assigned Numbers" [40].

In addition, for Agent Discovery, Mobile IP makes use of the existing Router Advertisement and Router Solicitation messages defined for ICMP Router Discovery [10].

Mobile IP defines a general Extension mechanism to allow optional information to be carried by Mobile IP control messages or by ICMP Router Discovery messages. Some extensions have been specified to be encoded in the simple Type-Length-Value format described in Section 1.9.

Extensions allow variable amounts of information to be carried within each datagram. The end of the list of Extensions is indicated by the total length of the IP datagram.

Two separately maintained sets of numbering spaces, from which Extension Type values are allocated, are used in Mobile IP:

- The first set consists of those Extensions which may appear only in Mobile IP control messages (those sent to and from UDP port number 434). In this document, the following Types are defined for Extensions appearing in Mobile IP control messages:

- 32 Mobile-Home Authentication
- 33 Mobile-Foreign Authentication
- 34 Foreign-Home Authentication

- The second set consists of those extensions which may appear only in ICMP Router Discovery messages [10]. In this document, the following Types are defined for Extensions appearing in ICMP Router Discovery messages:

- 0 One-byte Padding (encoded with no Length nor Data field)
- 16 Mobility Agent Advertisement
- 19 Prefix-Lengths

Each individual Extension is described in detail in a separate section later in this document. Up-to-date values for these Extension Type numbers are specified in the most recent "Assigned Numbers" [40].

Due to the separation (orthogonality) of these sets, it is conceivable that two Extensions that are defined at a later date could have identical Type values, so long as one of the Extensions may be used only in Mobile IP control messages and the other may be used only in ICMP Router Discovery messages.

The type field in the Mobile IP extension structure can support up to 255 (skippable and not skippable) uniquely identifiable extensions. When an Extension numbered in either of these sets within the range 0 through 127 is encountered but not recognized, the message containing that Extension **MUST** be silently discarded. When an Extension numbered in the range 128 through 255 is encountered which is not recognized, that particular Extension is ignored, but the rest of the Extensions and message data **MUST** still be processed. The Length field of the Extension is used to skip the Data field in searching for the next Extension.

Unless additional structure is utilized for the extension types, new developments or additions to Mobile IP might require so many new extensions that the available space for extension types might run out. Two new extension structures are proposed to solve this problem. Certain types of extensions can be aggregated, using

subtypes to identify the precise extension, for example as has been done with the Generic Authentication Keys extensions [35]. In many cases, this may reduce the rate of allocation for new values of the type field.

Since the new extension structures will cause an efficient usage of the extension type space, it is recommended that new Mobile IP extensions follow one of the two new extension formats whenever there may be the possibility to group related extensions together.

The following subsections provide details about three distinct structures for Mobile IP extensions:

- The simple extension format
- The long extension format
- The short extension format

1.9. Type-Length-Value Extension Format for Mobile IP Extensions

The Type-Length-Value format illustrated in figure 2 is used for extensions which are specified in this document. Since this simple extension structure does not encourage the most efficient usage of the extension type space, it is recommended that new Mobile IP extensions follow one of the two new extension formats specified in sections 1.10 or 1.11 whenever there may be the possibility to group related extensions together.

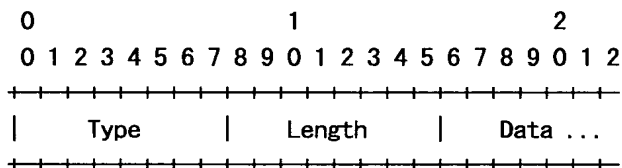
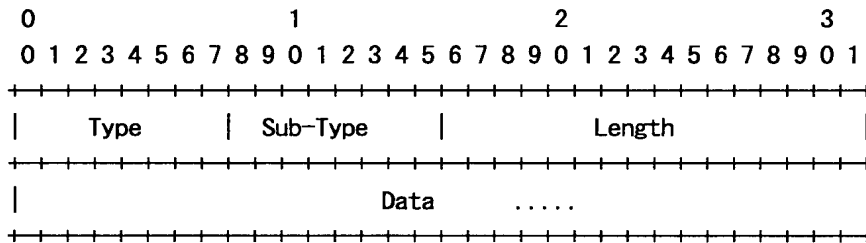


Figure 2: Type-Length-Value extension format for Mobile IPv4

- Type** Indicates the particular type of Extension.
- Length** Indicates the length (in bytes) of the data field within this Extension. The length does NOT include the Type and Length bytes.
- Data** The particular data associated with this Extension. This field may be zero or more bytes in length. The format and length of the data field is determined by the type and length fields.

1.10. Long Extension Format

This format is applicable for non-skippable extensions which carry information more than 256 bytes.



The Long Extension format requires that the following fields be specified as the first fields of the extension.

Type is the type, which describes a collection of extensions having a common data type.

Sub-Type is a unique number given to each member in the aggregated type.

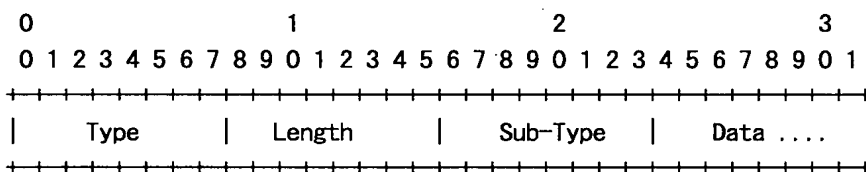
Length indicates the length (in bytes) of the data field within this Extension. It does NOT include the Type, Length and Sub-Type bytes.

Data is the data associated with the subtype of this extension. This specification does not place any additional structure on the subtype data.

Since the length field is 16 bits wide, the extension data can exceed 256 bytes in length.

1.11. Short Extension Format

This format is compatible with the skippable extensions defined in section 1.9. It is not applicable for extensions which require more than 256 bytes of data; for such extensions, use the format described in section 1.10.



The Short Extension format requires that the following fields be specified as the first fields of the extension:

Type is the type, which describes a collection of extensions having a common data type.

Sub-Type is a unique number given to each member in the aggregated type.

Length 8-bit unsigned integer. Length of the extension, in bytes, excluding the extension Type and the extension Length fields. This field **MUST** be set to 1 plus the total length of the data field.

Data is the data associated with this extension. This specification does not place any additional structure on the subtype data.

2. Agent Discovery

Agent Discovery is the method by which a mobile node determines whether it is currently connected to its home network or to a foreign network, and by which a mobile node can detect when it has moved from one network to another. When connected to a foreign network, the methods specified in this section also allow the mobile node to determine the foreign agent care-of address being offered by each foreign agent on that network.

Mobile IP extends ICMP Router Discovery [10] as its primary mechanism for Agent Discovery. An Agent Advertisement is formed by including a Mobility Agent Advertisement Extension in an ICMP Router Advertisement message (Section 2.1). An Agent Solicitation message is identical to an ICMP Router Solicitation, except that its IP TTL **MUST** be set to 1 (Section 2.2). This section describes the message formats and procedures by which mobile nodes, foreign agents, and home agents cooperate to realize Agent Discovery.

Agent Advertisement and Agent Solicitation may not be necessary for link layers that already provide this functionality. The method by which mobile nodes establish link-layer connections with prospective agents is outside the scope of this document (but see Appendix B). The procedures described below assume that such link-layer connectivity has already been established.

No authentication is required for Agent Advertisement and Agent Solicitation messages. They **MAY** be authenticated using the IP Authentication Header [22], which is unrelated to the messages described in this document. Further specification of the way in which Advertisement and Solicitation messages may be authenticated is outside of the scope of this document.

2.1. Agent Advertisement

Agent Advertisements are transmitted by a mobility agent to advertise its services on a link. Mobile nodes use these advertisements to determine their current point of attachment to the Internet. An Agent Advertisement is an ICMP Router Advertisement that has been extended to also carry an Mobility Agent Advertisement Extension (Section 2.1.1) and, optionally, a Prefix-Lengths Extension (Section 2.1.2), One-byte Padding Extension (Section 2.1.3), or other Extensions that might be defined in the future.

Within an Agent Advertisement message, ICMP Router Advertisement fields of the message are required to conform to the following additional specifications:

- Link-Layer Fields

Destination Address

The link-layer destination address of a unicast Agent Advertisement **MUST** be the same as the source link-layer address of the Agent Solicitation which prompted the Advertisement.

- IP Fields

TTL The TTL for all Agent Advertisements **MUST** be set to 1.

Destination Address

As specified for ICMP Router Discovery [10], the IP destination address of an multicast Agent Advertisement **MUST** be either the "all systems on this link" multicast address (224.0.0.1) [11] or the "limited broadcast" address (255.255.255.255). The subnet-directed broadcast address of the form <prefix>.<-1> cannot be used since mobile nodes will not generally know the prefix of the foreign network. When the Agent Advertisement is unicast to a mobile node, the IP home address of the mobile node **SHOULD** be used as the Destination Address.

– ICMP Fields

Code The Code field of the agent advertisement is interpreted as follows:

0 The mobility agent handles common traffic — that is, it acts as a router for IP datagrams not necessarily related to mobile nodes.

16 The mobility agent does not route common traffic. However, all foreign agents **MUST** (minimally) forward to a default router any datagrams received from a registered mobile node (Section 4.2.2).

Lifetime

The maximum length of time that the Advertisement is considered valid in the absence of further Advertisements.

Router Address(es)

See Section 2.3.1 for a discussion of the addresses that may appear in this portion of the Agent Advertisement.

Num Addr

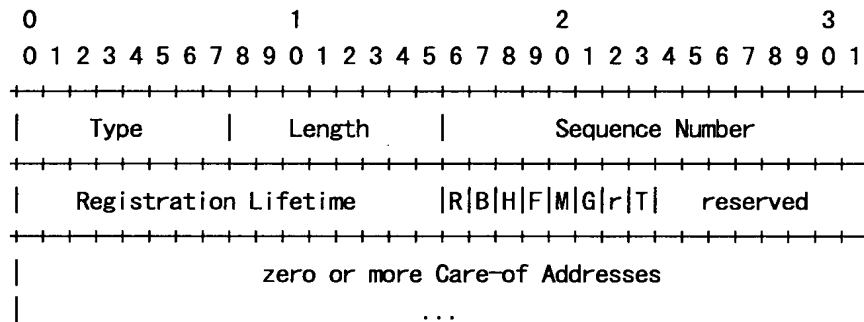
The number of Router Addresses advertised in this message. Note that in an Agent Advertisement message, the number of router addresses specified in the ICMP Router Advertisement portion of the message **MAY** be set to 0. See Section 2.3.1 for details.

If sent periodically, the nominal interval at which Agent Advertisements are sent **SHOULD** be no longer than 1/3 of the advertisement Lifetime given in the ICMP header. This interval **MAY** be shorter than 1/3 the advertised Lifetime. This allows a mobile node to miss three successive advertisements before deleting the agent from its list of valid agents. The actual transmission time for each advertisement **SHOULD** be slightly randomized [10] in order to avoid synchronization and subsequent collisions with other Agent

Advertisements that may be sent by other agents (or with other Router Advertisements sent by other routers). Note that this field has no relation to the "Registration Lifetime" field within the Mobility Agent Advertisement Extension defined below.

2.1.1. Mobility Agent Advertisement Extension

The Mobility Agent Advertisement Extension follows the ICMP Router Advertisement fields. It is used to indicate that an ICMP Router Advertisement message is also an Agent Advertisement being sent by a mobility agent. The Mobility Agent Advertisement Extension is defined as follows:



Type 16

Length (6 + 4*N), where 6 accounts for the number of bytes in the Sequence Number, Registration Lifetime, flags, and reserved fields, and N is the number of care-of addresses advertised.

Sequence Number

The count of Agent Advertisement messages sent since the agent was initialized (Section 2.3.2).

Registration Lifetime

The longest lifetime (measured in seconds) that this agent is willing to accept in any Registration Request. A value of 0xffff indicates infinity. This field has no relation to the "Lifetime" field within the ICMP Router Advertisement portion of the Agent Advertisement.

- R Registration required. Registration with this foreign agent (or another foreign agent on this link) is required even when using a co-located care-of address.
- B Busy. The foreign agent will not accept registrations from additional mobile nodes.
- H Home agent. This agent offers service as a home agent on the link on which this Agent Advertisement message is sent.

- F Foreign agent. This agent offers service as a foreign agent on the link on which this Agent Advertisement message is sent.
- M Minimal encapsulation. This agent implements receiving tunneled datagrams that use minimal encapsulation [34].
- G GRE encapsulation. This agent implements receiving tunneled datagrams that use GRE encapsulation [16].
- r Sent as zero; ignored on reception. SHOULD NOT be allocated for any other uses.
- T Foreign agent supports reverse tunneling [27].

reserved

Sent as zero; ignored on reception.

Care-of Address(es)

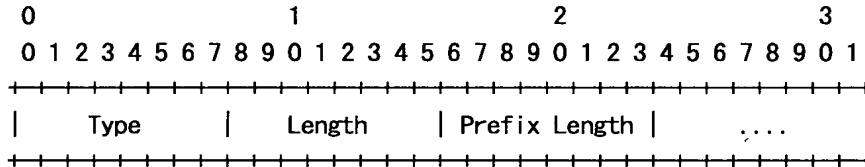
The advertised foreign agent care-of address(es) provided by this foreign agent. An Agent Advertisement MUST include at least one care-of address if the 'F' bit is set. The number of care-of addresses present is determined by the Length field in the Extension.

A home agent MUST always be prepared to serve the mobile nodes for which it is the home agent. A foreign agent may at times be too busy to serve additional mobile nodes; even so, it must continue to send Agent Advertisements, so that any mobile nodes already registered with it will know that they have not moved out of range of the foreign agent and that the foreign agent has not failed. A foreign agent may indicate that it is "too busy" to allow new mobile nodes to register with it, by setting the 'B' bit in its Agent Advertisements. An Agent Advertisement message MUST NOT have the 'B' bit set if the 'F' bit is not also set. Furthermore, at least one of the 'F' bit and the 'H' bit MUST be set in any Agent Advertisement message sent.

When a foreign agent wishes to require registration even from those mobile nodes which have acquired a co-located care-of address, it sets the 'R' bit to one. Because this bit applies only to foreign agents, an agent MUST NOT set the 'R' bit to one unless the 'F' bit is also set to one.

2.1.2. Prefix-Lengths Extension

The Prefix-Lengths Extension MAY follow the Mobility Agent Advertisement Extension. It is used to indicate the number of bits of network prefix that applies to each Router Address listed in the ICMP Router Advertisement portion of the Agent Advertisement. Note that the prefix lengths given DO NOT apply to care-of address(es) listed in the Mobility Agent Advertisement Extension. The Prefix-Lengths Extension is defined as follows:



Type 19 (Prefix-Lengths Extension)

Length N, where N is the value (possibly zero) of the Num Addrs field in the ICMP Router Advertisement portion of the Agent Advertisement.

Prefix Length(s)

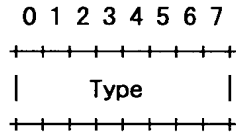
The number of leading bits that define the network number of the corresponding Router Address listed in the ICMP Router Advertisement portion of the message. The prefix length for each Router Address is encoded as a separate byte, in the order that the Router Addresses are listed in the ICMP Router Advertisement portion of the message.

See Section 2.4.2 for information about how the Prefix-Lengths Extension MAY be used by a mobile node when determining whether it has moved. See Appendix E for implementation details about the use of this Extension.

2.1.3. One-byte Padding Extension

Some IP protocol implementations insist upon padding ICMP messages to an even number of bytes. If the ICMP length of an Agent Advertisement is odd, this Extension MAY be included in order to make the ICMP length even. Note that this Extension is NOT intended to be a general-purpose Extension to be included in order to word- or long-align the various fields of the Agent Advertisement. An Agent Advertisement SHOULD NOT include more than one One-byte Padding Extension and if present, this Extension SHOULD be the last Extension in the Agent Advertisement.

Note that unlike other Extensions used in Mobile IP, the One-byte Padding Extension is encoded as a single byte, with no "Length" nor "Data" field present. The One-byte Padding Extension is defined as follows:



Type 0 (One-byte Padding Extension)

2.2. Agent Solicitation

An Agent Solicitation is identical to an ICMP Router Solicitation with the further restriction that the IP TTL Field **MUST** be set to 1.

2.3. Foreign Agent and Home Agent Considerations

Any mobility agent which cannot be discovered by a link-layer protocol **MUST** send Agent Advertisements. An agent which can be discovered by a link-layer protocol **SHOULD** also implement Agent Advertisements. However, the Advertisements need not be sent, except when the site policy requires registration with the agent (i.e., when the 'R' bit is set), or as a response to a specific Agent Solicitation. All mobility agents **MUST** process packets that they receive addressed to the Mobile-Agents multicast group, at address 224.0.0.11. A mobile node **MAY** send an Agent Solicitation to 224.0.0.11. All mobility agents **SHOULD** respond to Agent Solicitations.

The same procedures, defaults, and constants are used in Agent Advertisement messages and Agent Solicitation messages as specified for ICMP Router Discovery [10], except that:

- a mobility agent **MUST** limit the rate at which it sends broadcast or multicast Agent Advertisements; the maximum rate **SHOULD** be chosen so that the Advertisements do not consume a significant amount of network bandwidth, **AND**
- a mobility agent that receives a Router Solicitation **MUST NOT** require that the IP Source Address is the address of a neighbor (i.e., an address that matches one of the router's own addresses on the arrival interface, under the subnet mask associated with that address of the router).
- a mobility agent **MAY** be configured to send Agent Advertisements only in response to an Agent Solicitation message.

If the home network is not a virtual network, then the home agent for any mobile node SHOULD be located on the link identified by the mobile node's home address, and Agent Advertisement messages sent by the home agent on this link MUST have the 'H' bit set. In this way, mobile nodes on their own home network will be able to determine that they are indeed at home. Any Agent Advertisement messages sent by the home agent on another link to which it may be attached (if it is a mobility agent serving more than one link), MUST NOT have the 'H' bit set, unless the home agent also serves as a home agent (to other mobile nodes) on that other link. A mobility agent MAY use different settings for each of the 'R', 'H', and 'F' bits on different network interfaces.

If the home network is a virtual network, the home network has no physical realization external to the home agent itself. In this case, there is no physical network link on which to send Agent Advertisement messages advertising the home agent. Mobile nodes for which this is the home network are always treated as being away from home.

On a particular subnet, either all mobility agents MUST include the Prefix-Lengths Extension or all of them MUST NOT include this Extension. Equivalently, it is prohibited for some agents on a given subnet to include the Extension but for others not to include it. Otherwise, one of the move detection algorithms designed for mobile nodes will not function properly (Section 2.4.2).

2.3.1. Advertised Router Addresses

The ICMP Router Advertisement portion of the Agent Advertisement MAY contain one or more router addresses. An agent SHOULD only put its own addresses, if any, in the advertisement. Whether or not its own address appears in the Router Addresses, a foreign agent MUST route datagrams it receives from registered mobile nodes (Section 4.2.2).

2.3.2. Sequence Numbers and Rollover Handling

The sequence number in Agent Advertisements ranges from 0 to 0xffff. After booting, an agent MUST use the number 0 for its first advertisement. Each subsequent advertisement MUST use the sequence number one greater, with the exception that the sequence number 0xffff MUST be followed by sequence number 256. In this way, mobile nodes can distinguish a reduction in the sequence number that occurs after a reboot from a reduction that results in rollover of the sequence number after it attains the value 0xffff.

2.4. Mobile Node Considerations

Every mobile node **MUST** implement Agent Solicitation. Solicitations **SHOULD** only be sent in the absence of Agent Advertisements and when a care-of address has not been determined through a link-layer protocol or other means. The mobile node uses the same procedures, defaults, and constants for Agent Solicitation as specified for ICMP Router Solicitation messages [10], except that the mobile node **MAY** solicit more often than once every three seconds, and that a mobile node that is currently not connected to any foreign agent **MAY** solicit more times than `MAX_SOLICITATIONS`.

The rate at which a mobile node sends Solicitations **MUST** be limited by the mobile node. The mobile node **MAY** send three initial Solicitations at a maximum rate of one per second while searching for an agent. After this, the rate at which Solicitations are sent **MUST** be reduced so as to limit the overhead on the local link. Subsequent Solicitations **MUST** be sent using a binary exponential backoff mechanism, doubling the interval between consecutive Solicitations, up to a maximum interval. The maximum interval **SHOULD** be chosen appropriately based upon the characteristics of the media over which the mobile node is soliciting. This maximum interval **SHOULD** be at least one minute between Solicitations.

While still searching for an agent, the mobile node **MUST NOT** increase the rate at which it sends Solicitations unless it has received a positive indication that it has moved to a new link. After successfully registering with an agent, the mobile node **SHOULD** also increase the rate at which it will send Solicitations when it next begins searching for a new agent with which to register. The increased solicitation rate **MAY** revert to the maximum rate, but then **MUST** be limited in the manner described above. In all cases, the recommended solicitation intervals are nominal values. Mobile nodes **MUST** randomize their solicitation times around these nominal values as specified for ICMP Router Discovery [10].

Mobile nodes **MUST** process received Agent Advertisements. A mobile node can distinguish an Agent Advertisement message from other uses of the ICMP Router Advertisement message by examining the number of advertised addresses and the IP Total Length field. When the IP total length indicates that the ICMP message is longer than needed for the number of advertised addresses, the remaining data is interpreted as one or more Extensions. The presence of a Mobility Agent Advertisement Extension identifies the advertisement as an Agent Advertisement.

If there is more than one advertised address, the mobile node SHOULD pick the first address for its initial registration attempt. If the registration attempt fails with a status Code indicating rejection by the foreign agent, the mobile node MAY retry the attempt with each subsequent advertised address in turn.

When multiple methods of agent discovery are in use, the mobile node SHOULD first attempt registration with agents including Mobility Agent Advertisement Extensions in their advertisements, in preference to those discovered by other means. This preference maximizes the likelihood that the registration will be recognized, thereby minimizing the number of registration attempts.

A mobile node MUST ignore reserved bits in Agent Advertisements, as opposed to discarding such advertisements. In this way, new bits can be defined later, without affecting the ability for mobile nodes to use the advertisements even when the newly defined bits are not understood.

2.4.1. Registration Required

When the mobile node receives an Agent Advertisement with the 'R' bit set, the mobile node SHOULD register through the foreign agent, even when the mobile node might be able to acquire its own co-located care-of address. This feature is intended to allow sites to enforce visiting policies (such as accounting) which require exchanges of authorization.

If formerly reserved bits require some kind of monitoring/enforcement at the foreign link, foreign agents implementing the new specification for the formerly reserved bits can set the 'R' bit. This has the effect of forcing the mobile node to register through the foreign agent, so the foreign agent could then monitor/enforce the policy.

2.4.2. Move Detection

Two primary mechanisms are provided for mobile nodes to detect when they have moved from one subnet to another. Other mechanisms MAY also be used. When the mobile node detects that it has moved, it SHOULD register (Section 3) with a suitable care-of address on the new foreign network. However, the mobile node MUST NOT register more frequently than once per second on average, as specified in Section 3.6.3.

2.4.2.1. Algorithm 1

The first method of move detection is based upon the Lifetime field within the main body of the ICMP Router Advertisement portion of the Agent Advertisement. A mobile node SHOULD record the Lifetime received in any Agent Advertisements, until that Lifetime expires. If the mobile node fails to receive another advertisement from the same agent within the specified Lifetime, it SHOULD assume that it has lost contact with that agent. If the mobile node has previously received an Agent Advertisement from another agent for which the Lifetime field has not yet expired, the mobile node MAY immediately attempt registration with that other agent. Otherwise, the mobile node SHOULD attempt to discover a new agent with which to register.

2.4.2.2. Algorithm 2

The second method uses network prefixes. The Prefix-Lengths Extension MAY be used in some cases by a mobile node to determine whether or not a newly received Agent Advertisement was received on the same subnet as the mobile node's current care-of address. If the prefixes differ, the mobile node MAY assume that it has moved. If a mobile node is currently using a foreign agent care-of address, the mobile node SHOULD NOT use this method of move detection unless both the current agent and the new agent include the Prefix-Lengths Extension in their respective Agent Advertisements; if this Extension is missing from one or both of the advertisements, this method of move detection SHOULD NOT be used. Similarly, if a mobile node is using a co-located care-of address, it SHOULD not use this method of move detection unless the new agent includes the Prefix-Lengths Extension in its Advertisement and the mobile node knows the network prefix of its current co-located care-of address. On the expiration of its current registration, if this method indicates that the mobile node has moved, rather than re-registering with its current care-of address, a mobile node MAY choose instead to register with a the foreign agent sending the new Advertisement with the different network prefix. The Agent Advertisement on which the new registration is based MUST NOT have expired according to its Lifetime field.

2.4.3. Returning Home

A mobile node can detect that it has returned to its home network when it receives an Agent Advertisement from its own home agent. If so, it SHOULD deregister with its home agent (Section 3). Before attempting to deregister, the mobile node SHOULD configure its routing table appropriately for its home network (Section 4.2.1). In

addition, if the home network is using ARP [36], the mobile node **MUST** follow the procedures described in Section 4.6 with regard to ARP, proxy ARP, and gratuitous ARP.

2.4.4. Sequence Numbers and Rollover Handling

If a mobile node detects two successive values of the sequence number in the Agent Advertisements from the foreign agent with which it is registered, the second of which is less than the first and inside the range 0 to 255, the mobile node **SHOULD** register again. If the second value is less than the first but is greater than or equal to 256, the mobile node **SHOULD** assume that the sequence number has rolled over past its maximum value (0xffff), and that reregistration is not necessary (Section 2.3).

3. Registration

Mobile IP registration provides a flexible mechanism for mobile nodes to communicate their current reachability information to their home agent. It is the method by which mobile nodes:

- request forwarding services when visiting a foreign network,
- inform their home agent of their current care-of address,
- renew a registration which is due to expire, and/or
- deregister when they return home.

Registration messages exchange information between a mobile node, (optionally) a foreign agent, and the home agent. Registration creates or modifies a mobility binding at the home agent, associating the mobile node's home address with its care-of address for the specified Lifetime.

Several other (optional) capabilities are available through the registration procedure, which enable a mobile node to:

- discover its home address, if the mobile node is not configured with this information.
- maintain multiple simultaneous registrations, so that a copy of each datagram will be tunneled to each active care-of address
- deregister specific care-of addresses while retaining other mobility bindings, and

- discover the address of a home agent if the mobile node is not configured with this information.

3.1. Registration Overview

Mobile IP defines two different registration procedures, one via a foreign agent that relays the registration to the mobile node's home agent, and one directly with the mobile node's home agent. The following rules determine which of these two registration procedures to use in any particular circumstance:

- If a mobile node is registering a foreign agent care-of address, the mobile node **MUST** register via that foreign agent.
- If a mobile node is using a co-located care-of address, and receives an Agent Advertisement from a foreign agent on the link on which it is using this care-of address, the mobile node **SHOULD** register via that foreign agent (or via another foreign agent on this link) if the 'R' bit is set in the received Agent Advertisement message.
- If a mobile node is otherwise using a co-located care-of address, the mobile node **MUST** register directly with its home agent.
- If a mobile node has returned to its home network and is (de)registering with its home agent, the mobile node **MUST** register directly with its home agent.

Both registration procedures involve the exchange of Registration Request and Registration Reply messages (Sections 3.3 and 3.4). When registering via a foreign agent, the registration procedure requires the following four messages:

- a) The mobile node sends a Registration Request to the prospective foreign agent to begin the registration process.
- b) The foreign agent processes the Registration Request and then relays it to the home agent.
- c) The home agent sends a Registration Reply to the foreign agent to grant or deny the Request.
- d) The foreign agent processes the Registration Reply and then relays it to the mobile node to inform it of the disposition of its Request.

When the mobile node instead registers directly with its home agent, the registration procedure requires only the following two messages:

- a) The mobile node sends a Registration Request to the home agent.
- b) The home agent sends a Registration Reply to the mobile node, granting or denying the Request.

The registration messages defined in Sections 3.3 and 3.4 use the User Datagram Protocol (UDP) [37]. A nonzero UDP checksum SHOULD be included in the header, and MUST be checked by the recipient. A zero UDP checksum SHOULD be accepted by the recipient. The behavior of the mobile node and the home agent with respect to their mutual acceptance of packets with zero UDP checksums SHOULD be defined as part of the mobility security association which exists between them.

3.2. Authentication

Each mobile node, foreign agent, and home agent MUST be able to support a mobility security association for mobile entities, indexed by their SPI and IP address. In the case of the mobile node, this must be its Home Address. See Section 5.1 for requirements for support of authentication algorithms. Registration messages between a mobile node and its home agent MUST be authenticated with an authorization-enabling extension, e.g. the Mobile-Home Authentication Extension (Section 3.5.2). This extension MUST be the first authentication extension; other foreign agent-specific extensions MAY be added to the message after the mobile node computes the authentication.

3.3. Registration Request

A mobile node registers with its home agent using a Registration Request message so that its home agent can create or modify a mobility binding for that mobile node (e.g., with a new lifetime). The Request may be relayed to the home agent by the foreign agent through which the mobile node is registering, or it may be sent directly to the home agent in the case in which the mobile node is registering a co-located care-of address.

IP fields:

Source Address Typically the interface address from which the message is sent.

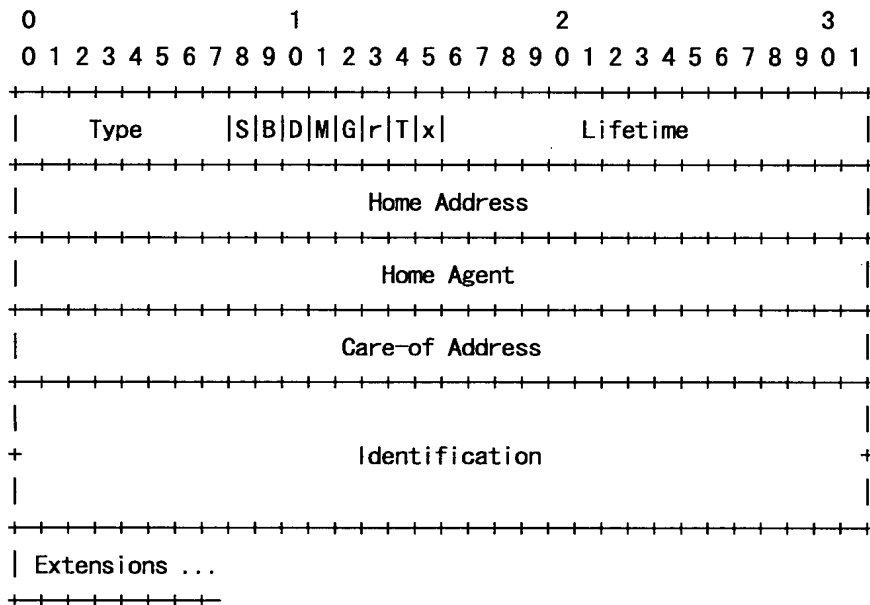
Destination Address Typically that of the foreign agent or the home agent.

See Sections 3.6.1.1 and 3.7.2.2 for details. UDP fields:

Source Port variable

Destination Port 434

The UDP header is followed by the Mobile IP fields shown below:



Type 1 (Registration Request)

- S** Simultaneous bindings. If the 'S' bit is set, the mobile node is requesting that the home agent retain its prior mobility bindings, as described in Section 3.6.1.2.
- B** Broadcast datagrams. If the 'B' bit is set, the mobile node requests that the home agent tunnel to it any broadcast datagrams that it receives on the home network, as described in Section 4.3.
- D** Decapsulation by mobile node. If the 'D' bit is set, the mobile node will itself decapsulate datagrams which are sent to the care-of address. That is, the mobile node is using a co-located care-of address.
- M** Minimal encapsulation. If the 'M' bit is set, the mobile node requests that its home agent use minimal encapsulation [34] for datagrams tunneled to the mobile node.

- G GRE encapsulation. If the 'G' bit is set, the mobile node requests that its home agent use GRE encapsulation [16] for datagrams tunneled to the mobile node.
- r Sent as zero; ignored on reception. SHOULD NOT be allocated for any other uses.
- T Reverse Tunneling requested; see [27].
- x Sent as zero; ignored on reception.

Lifetime

The number of seconds remaining before the registration is considered expired. A value of zero indicates a request for deregistration. A value of 0xffff indicates infinity.

Home Address

The IP address of the mobile node.

Home Agent

The IP address of the mobile node's home agent.

Care-of Address

The IP address for the end of the tunnel.

Identification

A 64-bit number, constructed by the mobile node, used for matching Registration Requests with Registration Replies, and for protecting against replay attacks of registration messages. See Sections 5.4 and 5.7.

Extensions

The fixed portion of the Registration Request is followed by one or more of the Extensions listed in Section 3.5. An authorization-enabling extension MUST be included in all Registration Requests. See Sections 3.6.1.3 and 3.7.2.2 for information on the relative order in which different extensions, when present, MUST be placed in a Registration Request message.

3.4. Registration Reply

A mobility agent returns a Registration Reply message to a mobile node which has sent a Registration Request (Section 3.3) message. If the mobile node is requesting service from a foreign agent, that foreign agent will receive the Reply from the home agent and subsequently relay it to the mobile node. The Reply message contains the necessary codes to inform the mobile node about the status of its Request, along with the lifetime granted by the home agent, which MAY be smaller than the original Request.

The foreign agent **MUST NOT** increase the Lifetime selected by the mobile node in the Registration Request, since the Lifetime is covered by an authentication extension which enables authorization by the home agent. Such an extension contains authentication data which cannot be correctly (re)computed by the foreign agent. The home agent **MUST NOT** increase the Lifetime selected by the mobile node in the Registration Request, since doing so could increase it beyond the maximum Registration Lifetime allowed by the foreign agent. If the Lifetime received in the Registration Reply is greater than that in the Registration Request, the Lifetime in the Request **MUST** be used. When the Lifetime received in the Registration Reply is less than that in the Registration Request, the Lifetime in the Reply **MUST** be used.

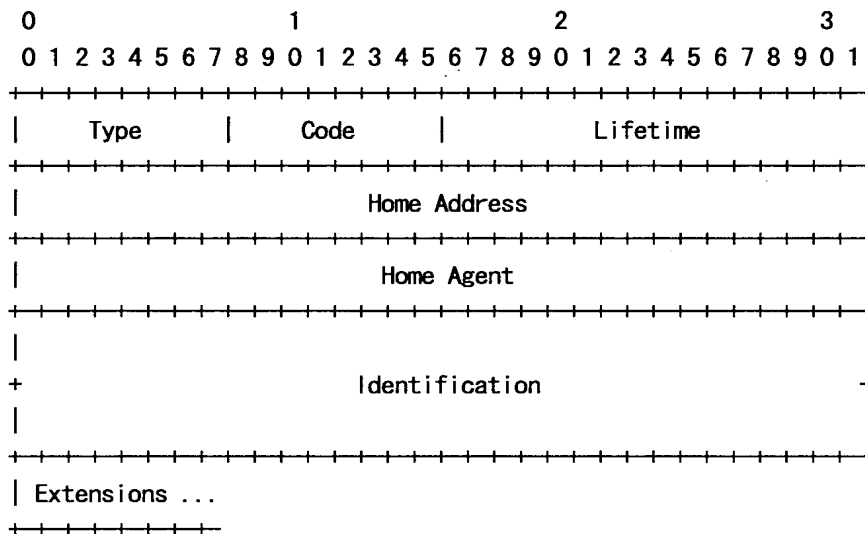
IP fields:

Source Address	Typically copied from the destination address of the Registration Request to which the agent is replying. See Sections 3.7.2.3 and 3.8.3.1 for complete details.
Destination Address	Copied from the source address of the Registration Request to which the agent is replying

UDP fields:

Source Port	<variable>
Destination Port	Copied from the source port of the corresponding Registration Request (Section 3.7.1).

The UDP header is followed by the Mobile IP fields shown below:



Type 3 (Registration Reply)

Code A value indicating the result of the Registration Request. See below for a list of currently defined Code values.

Lifetime

If the Code field indicates that the registration was accepted, the Lifetime field is set to the number of seconds remaining before the registration is considered expired. A value of zero indicates that the mobile node has been deregistered. A value of 0xffff indicates infinity. If the Code field indicates that the registration was denied, the contents of the Lifetime field are unspecified and MUST be ignored on reception.

Home Address

The IP address of the mobile node.

Home Agent

The IP address of the mobile node's home agent.

Identification

A 64-bit number used for matching Registration Requests with Registration Replies, and for protecting against replay attacks of registration messages. The value is

based on the Identification field from the Registration Request message from the mobile node, and on the style of replay protection used in the security context between the mobile node and its home agent (defined by the mobility security association between them, and SPI value in the authorization-enabling extension). See Sections 5.4 and 5.7.

Extensions

The fixed portion of the Registration Reply is followed by one or more of the Extensions listed in Section 3.5. An authorization-enabling extension **MUST** be included in all Registration Replies returned by the home agent. See Sections 3.7.2.2 and 3.8.3.3 for rules on placement of extensions to Reply messages.

The following values are defined for use within the Code field.

Registration successful:

- 0 registration accepted
- 1 registration accepted, but simultaneous mobility bindings unsupported

Registration denied by the foreign agent:

- 64 reason unspecified
- 65 administratively prohibited
- 66 insufficient resources
- 67 mobile node failed authentication
- 68 home agent failed authentication
- 69 requested Lifetime too long
- 70 poorly formed Request
- 71 poorly formed Reply
- 72 requested encapsulation unavailable
- 73 reserved and unavailable
- 77 invalid care-of address
- 78 registration timeout
- 80 home network unreachable (ICMP error received)
- 81 home agent host unreachable (ICMP error received)
- 82 home agent port unreachable (ICMP error received)
- 88 home agent unreachable (other ICMP error received)

Registration denied by the home agent:

- 128 reason unspecified
- 129 administratively prohibited
- 130 insufficient resources
- 131 mobile node failed authentication
- 132 foreign agent failed authentication
- 133 registration Identification mismatch
- 134 poorly formed Request
- 135 too many simultaneous mobility bindings
- 136 unknown home agent address

Up-to-date values of the Code field are specified in the most recent "Assigned Numbers" [40].

3.5. Registration Extensions

3.5.1. Computing Authentication Extension Values

The Authenticator value computed for each authentication Extension MUST protect the following fields from the registration message:

- the UDP payload (that is, the Registration Request or Registration Reply data),
- all prior Extensions in their entirety, and
- the Type, Length, and SPI of this Extension.

The default authentication algorithm uses HMAC-MD5 [23] to compute a 128-bit "message digest" of the registration message. The data over which the HMAC is computed is defined as:

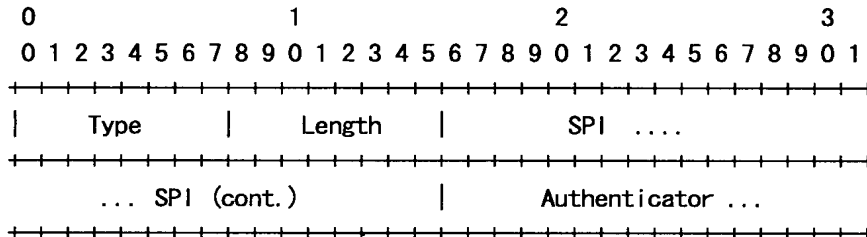
- the UDP payload (that is, the Registration Request or Registration Reply data),
- all prior Extensions in their entirety, and
- the Type, Length, and SPI of this Extension.

Note that the Authenticator field itself and the UDP header are NOT included in the computation of the default Authenticator value. See Section 5.1 for information about support requirements for message authentication codes, which are to be used with the various authentication Extensions.

The Security Parameter Index (SPI) within any of the authentication Extensions defines the security context which is used to compute the Authenticator value and which MUST be used by the receiver to check that value. In particular, the SPI selects the authentication algorithm and mode (Section 5.1) and secret (a shared key, or appropriate public/private key pair) used in computing the Authenticator. In order to ensure interoperability between different implementations of the Mobile IP protocol, an implementation MUST be able to associate any SPI value with any authentication algorithm and mode which it implements. In addition, all implementations of Mobile IP MUST implement the default authentication algorithm (HMAC-MD5) specified above.

3.5.2. Mobile-Home Authentication Extension

Exactly one authorization-enabling extension MUST be present in all Registration Requests, and also in all Registration Replies generated by the Home Agent. The Mobile-Home Authentication Extension is always an authorization-enabling for registration messages specified in this document. This requirement is intended to eliminate problems [2] which result from the uncontrolled propagation of remote redirects in the Internet. The location of the extension marks the end of the authenticated data.



Type 32

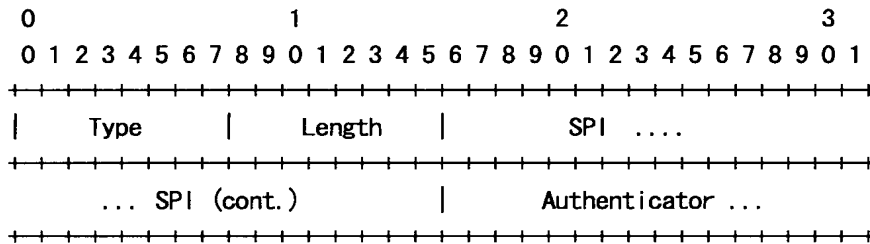
Length 4 plus the number of bytes in the Authenticator.

SPI Security Parameter Index (4 bytes). An opaque identifier (see Section 1.6).

Authenticator (variable length) (See Section 3.5.1.)

3.5.3. Mobile-Foreign Authentication Extension

This Extension MAY be included in Registration Requests and Replies in cases in which a mobility security association exists between the mobile node and the foreign agent. See Section 5.1 for information about support requirements for message authentication codes.



Type 33

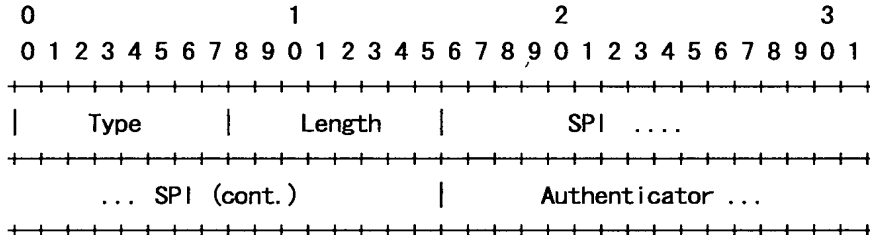
Length 4 plus the number of bytes in the Authenticator.

SPI Security Parameter Index (4 bytes). An opaque identifier (see Section 1.6).

Authenticator (variable length) (See Section 3.5.1.)

3.5.4. Foreign-Home Authentication Extension

This Extension MAY be included in Registration Requests and Replies in cases in which a mobility security association exists between the foreign agent and the home agent. See Section 5.1 for information about support requirements for message authentication codes.



Type 34

Length 4 plus the number of bytes in the Authenticator.

SPI Security Parameter Index (4 bytes). An opaque identifier (see Section 1.6).

Authenticator (variable length) (See Section 3.5.1.)

3.6. Mobile Node Considerations

A mobile node MUST be configured with a netmask and a mobility security association for each of its home agents. In addition, a mobile node MAY be configured with its home address, and the IP

address of one or more of its home agents; otherwise, the mobile node MAY discover a home agent using the procedures described in Section 3.6.1.2.

If the mobile node is not configured with a home address, it MAY use the Mobile Node NAI extension [6] to identify itself, and set the Home Address field of the Registration Request to 0.0.0.0. In this case, the mobile node MUST be able to assign its home address after extracting this information from the Registration Reply from the home agent.

For each pending registration, the mobile node maintains the following information:

- the link-layer address of the foreign agent to which the Registration Request was sent, if applicable,
- the IP destination address of the Registration Request,
- the care-of address used in the registration,
- the Identification value sent in the registration,
- the originally requested Lifetime, and
- the remaining Lifetime of the pending registration.

A mobile node SHOULD initiate a registration whenever it detects a change in its network connectivity. See Section 2.4.2 for methods by which mobile nodes MAY make such a determination. When it is away from home, the mobile node's Registration Request allows its home agent to create or modify a mobility binding for it. When it is at home, the mobile node's (de)Registration Request allows its home agent to delete any previous mobility binding(s) for it. A mobile node operates without the support of mobility functions when it is at home.

There are other conditions under which the mobile node SHOULD (re)register with its foreign agent, such as when the mobile node detects that the foreign agent has rebooted (as specified in Section 2.4.4) and when the current registration's Lifetime is near expiration.

In the absence of link-layer indications of changes in point of attachment, Agent Advertisements from new agents SHOULD NOT cause a mobile node to attempt a new registration, if its current registration has not expired and it is still also receiving Agent Advertisements from the foreign agent with which it is currently registered. In the absence of link-layer indications, a mobile node MUST NOT attempt to register more often than once per second.

A mobile node MAY register with a different agent when transport-layer protocols indicate excessive retransmissions. A mobile node MUST NOT consider reception of an ICMP Redirect from a foreign agent that is currently providing service to it as reason to register with a new foreign agent. Within these constraints, the mobile node MAY register again at any time.

Appendix D shows some examples of how the fields in registration messages would be set up in some typical registration scenarios.

3.6.1. Sending Registration Requests

The following sections specify details for the values the mobile node MUST supply in the fields of Registration Request messages.

3.6.1.1. IP Fields

This section provides the specific rules by which mobile nodes pick values for the IP header fields of a Registration Request.

IP Source Address:

- When registering on a foreign network with a co-located care-of address, the IP source address MUST be the care-of address.
- Otherwise, if the mobile node does not have a home address, the IP source address MUST be 0.0.0.0.
- In all other circumstances, the IP source address MUST be the mobile node's home address.

IP Destination Address:

- When the mobile node has discovered the agent with which it is registering, through some means (e.g., link-layer) that does not provide the IP address of the agent (the IP address of the agent is unknown to the mobile node), then the "All Mobility Agents" multicast address (224.0.0.11) MUST be used. In this case, the mobile node MUST use the agent's link-layer unicast address in order to deliver the datagram to the correct agent.
- When registering with a foreign agent, the address of the agent as learned from the IP source address of the corresponding Agent Advertisement MUST be used. This MAY be an address which does not appear as an advertised care-of address in the Agent Advertisement. In addition, when transmitting this Registration Request message, the mobile node MUST use a link-

layer destination address copied from the link-layer source address of the Agent Advertisement message in which it learned this foreign agent's IP address.

- When the mobile node is registering directly with its home agent and knows the (unicast) IP address of its home agent, the destination address **MUST** be set to this address.
- If the mobile node is registering directly with its home agent, but does not know the IP address of its home agent, the mobile node may use dynamic home agent address resolution to automatically determine the IP address of its home agent (Section 3.6.1.2). In this case, the IP destination address is set to the subnet-directed broadcast address of the mobile node's home network. This address **MUST NOT** be used as the destination IP address if the mobile node is registering via a foreign agent, although it **MAY** be used as the Home Agent address in the body of the Registration Request when registering via a foreign agent.

IP Time to Live:

- The IP TTL field **MUST** be set to 1 if the IP destination address is set to the "All Mobility Agents" multicast address as described above. Otherwise a suitable value should be chosen in accordance with standard IP practice [38].

3.6.1.2. Registration Request Fields

This section provides specific rules by which mobile nodes pick values for the fields within the fixed portion of a Registration Request.

A mobile node **MAY** set the 'S' bit in order to request that the home agent maintain prior mobility binding(s). Otherwise, the home agent deletes any previous binding(s) and replaces them with the new binding specified in the Registration Request. Multiple simultaneous mobility bindings are likely to be useful when a mobile node using at least one wireless network interface moves within wireless transmission range of more than one foreign agent. IP explicitly allows duplication of datagrams. When the home agent allows simultaneous bindings, it will tunnel a separate copy of each arriving datagram to each care-of address, and the mobile node will receive multiple copies of datagrams destined to it.

The mobile node **SHOULD** set the 'D' bit if it is registering with a co-located care-of address. Otherwise, the 'D' bit **MUST NOT** be set.

A mobile node MAY set the 'B' bit to request its home agent to forward to it, a copy of broadcast datagrams received by its home agent from the home network. The method used by the home agent to forward broadcast datagrams depends on the type of care-of address registered by the mobile node, as determined by the 'D' bit in the mobile node's Registration Request:

- If the 'D' bit is set, then the mobile node has indicated that it will decapsulate any datagrams tunneled to this care-of address itself (the mobile node is using a co-located care-of address). In this case, to forward such a received broadcast datagram to the mobile node, the home agent MUST tunnel it to this care-of address. The mobile node de-tunnels the received datagram in the same way as any other datagram tunneled directly to it.
- If the 'D' bit is NOT set, then the mobile node has indicated that it is using a foreign agent care-of address, and that the foreign agent will thus decapsulate arriving datagrams before forwarding them to the mobile node. In this case, to forward such a received broadcast datagram to the mobile node, the home agent MUST first encapsulate the broadcast datagram in a unicast datagram addressed to the mobile node's home address, and then MUST tunnel this resulting datagram to the mobile node's care-of address.

When decapsulated by the foreign agent, the inner datagram will thus be a unicast IP datagram addressed to the mobile node, identifying to the foreign agent the intended destination of the encapsulated broadcast datagram, and will be delivered to the mobile node in the same way as any tunneled datagram arriving for the mobile node. The foreign agent MUST NOT decapsulate the encapsulated broadcast datagram and MUST NOT use a local network broadcast to transmit it to the mobile node. The mobile node thus MUST decapsulate the encapsulated broadcast datagram itself, and thus MUST NOT set the 'B' bit in its Registration Request in this case unless it is capable of decapsulating datagrams.

The mobile node MAY request alternative forms of encapsulation by setting the 'M' bit and/or the 'G' bit, but only if the mobile node is decapsulating its own datagrams (the mobile node is using a co-located care-of address) or if its foreign agent has indicated support for these forms of encapsulation by setting the corresponding bits in the Mobility Agent Advertisement Extension of an Agent Advertisement received by the mobile node. Otherwise, the mobile node MUST NOT set these bits.

The Lifetime field is chosen as follows:

- If the mobile node is registering with a foreign agent, the Lifetime SHOULD NOT exceed the value in the Registration Lifetime field of the Agent Advertisement message received from the foreign agent.
When the method by which the care-of address is learned does not include a Lifetime, the default ICMP Router Advertisement Lifetime (1800 seconds) MAY be used.
- The mobile node MAY ask a home agent to delete a particular mobility binding, by sending a Registration Request with the care-of address for this binding, with the Lifetime field set to zero (Section 3.8.2).
- Similarly, a Lifetime of zero is used when the mobile node deregisters all care-of addresses, such as upon returning home.

The Home Address field MUST be set to the mobile node's home address, if this information is known. Otherwise, the Home Address MUST be set to zeroes.

The Home Agent field MUST be set to the address of the mobile node's home agent, if the mobile node knows this address. Otherwise, the mobile node MAY use dynamic home agent address resolution to learn the address of its home agent. In this case, the mobile node MUST set the Home Agent field to the subnet-directed broadcast address of the mobile node's home network. Each home agent receiving such a Registration Request with a broadcast destination address MUST reject the mobile node's registration and SHOULD return a rejection Registration Reply indicating its unicast IP address for use by the mobile node in a future registration attempt.

The Care-of Address field MUST be set to the value of the particular care-of address that the mobile node wishes to (de)register. In the special case in which a mobile node wishes to deregister all care-of addresses, it MUST set this field to its home address.

The mobile node chooses the Identification field in accordance with the style of replay protection it uses with its home agent. This is part of the mobility security association the mobile node shares with its home agent. See Section 5.7 for the method by which the mobile node computes the Identification field.

3.6.1.3. Extensions

This section describes the ordering of any mandatory and any optional Extensions that a mobile node appends to a Registration Request.

This following ordering **MUST** be followed:

- a) The IP header, followed by the UDP header, followed by the fixed-length portion of the Registration Request, followed by
- b) If present, any non-authentication Extensions expected to be used by the home agent (which may or may not also be useful to the foreign agent), followed by
- c) An authorization-enabling extension, followed by
- d) If present, any non-authentication Extensions used only by the foreign agent, followed by
- e) The Mobile-Foreign Authentication Extension, if present.

Note that items (a) and (c) **MUST** appear in every Registration Request sent by the mobile node. Items (b), (d), and (e) are optional. However, item (e) **MUST** be included when the mobile node and the foreign agent share a mobility security association.

3.6.2. Receiving Registration Replies

Registration Replies will be received by the mobile node in response to its Registration Requests. Registration Replies generally fall into three categories:

- the registration was accepted,
- the registration was denied by the foreign agent, or
- the registration was denied by the home agent.

The remainder of this section describes the Registration Reply handling by a mobile node in each of these three categories.

3.6.2.1. Validity Checks

Registration Replies with an invalid, non-zero UDP checksum **MUST** be silently discarded.

In addition, the low-order 32 bits of the Identification field in the Registration Reply **MUST** be compared to the low-order 32 bits of the Identification field in the most recent Registration Request sent to the replying agent. If they do not match, the Reply **MUST** be silently discarded.

Also, the Registration Reply MUST be checked for presence of an authorization-enabling extension. For all Registration Reply messages containing a Status Code indicating status from the Home Agent, the mobile node MUST check for the presence of an authorization-enabling extension, acting in accordance with the Code field in the Reply. The rules are as follows:

- a) If the mobile node and the foreign agent share a mobility security association, exactly one Mobile-Foreign Authentication Extension MUST be present in the Registration Reply, and the mobile node MUST check the Authenticator value in the Extension. If no Mobile-Foreign Authentication Extension is found, or if more than one Mobile-Foreign Authentication Extension is found, or if the Authenticator is invalid, the mobile node MUST silently discard the Reply and SHOULD log the event as a security exception.
- b) If the Code field indicates that service is denied by the home agent, or if the Code field indicates that the registration was accepted by the home agent, exactly one Mobile-Home Authentication Extension MUST be present in the Registration Reply, and the mobile node MUST check the Authenticator value in the Extension. If the Registration Reply was generated by the home agent but no Mobile-Home Authentication Extension is found, or if more than one Mobile-Home Authentication Extension is found, or if the Authenticator is invalid, the mobile node MUST silently discard the Reply and SHOULD log the event as a security exception.

If the Code field indicates an authentication failure, either at the foreign agent or the home agent, then it is quite possible that any authenticators in the Registration Reply will also be in error. This could happen, for example, if the shared secret between the mobile node and home agent was erroneously configured. The mobile node SHOULD log such errors as security exceptions.

3.6.2.2. Registration Request Accepted

If the Code field indicates that the request has been accepted, the mobile node SHOULD configure its routing table appropriately for its current point of attachment (Section 4.2.1).

If the mobile node is returning to its home network and that network is one which implements ARP, the mobile node MUST follow the procedures described in Section 4.6 with regard to ARP, proxy ARP, and gratuitous ARP.

If the mobile node has registered on a foreign network, it SHOULD re-register before the expiration of the Lifetime of its registration. As described in Section 3.6, for each pending Registration Request, the mobile node MUST maintain the remaining lifetime of this pending registration, as well as the original Lifetime from the Registration Request. When the mobile node receives a valid Registration Reply, the mobile node MUST decrease its view of the remaining lifetime of the registration by the amount by which the home agent decreased the originally requested Lifetime. This procedure is equivalent to the mobile node starting a timer for the granted Lifetime at the time it sent the Registration Request, even though the granted Lifetime is not known to the mobile node until the Registration Reply is received. Since the Registration Request is certainly sent before the home agent begins timing the registration Lifetime (also based on the granted Lifetime), this procedure ensures that the mobile node will re-register before the home agent expires and deletes the registration, in spite of possibly non-negligible transmission delays for the original Registration Request and Reply that started the timing of the Lifetime at the mobile node and its home agent.

3.6.2.3. Registration Request Denied

If the Code field indicates that service is being denied, the mobile node SHOULD log the error. In certain cases the mobile node may be able to "repair" the error. These include:

Code 69: (Denied by foreign agent, Lifetime too long)

In this case, the Lifetime field in the Registration Reply will contain the maximum Lifetime value which that foreign agent is willing to accept in any Registration Request. The mobile node MAY attempt to register with this same agent, using a Lifetime in the Registration Request that MUST be less than or equal to the value specified in the Reply.

Code 133: (Denied by home agent, Identification mismatch)

In this case, the Identification field in the Registration Reply will contain a value that allows the mobile node to synchronize with the home agent, based upon the style of replay protection in effect (Section 5.7). The mobile node MUST adjust the parameters it uses to compute the Identification field based upon the information in the Registration Reply, before issuing any future Registration Requests.

Code 136: (Denied by home agent, Unknown home agent address)

This code is returned by a home agent when the mobile node is performing dynamic home agent address resolution as described in Sections 3.6.1.1 and 3.6.1.2. In this case, the Home Agent field within the Reply will contain the unicast IP address of the home agent returning the Reply. The mobile node MAY then attempt to register with this home agent in future Registration Requests. In addition, the mobile node SHOULD adjust the parameters it uses to compute the Identification field based upon the corresponding field in the Registration Reply, before issuing any future Registration Requests.

3.6.3. Registration Retransmission

When no Registration Reply has been received within a reasonable time, another Registration Request MAY be transmitted. When timestamps are used, a new registration Identification is chosen for each retransmission; thus it counts as a new registration. When nonces are used, the unanswered Request is retransmitted unchanged; thus the retransmission does not count as a new registration (Section 5.7). In this way a retransmission will not require the home agent to resynchronize with the mobile node by issuing another nonce in the case in which the original Registration Request (rather than its Registration Reply) was lost by the network.

The maximum time until a new Registration Request is sent SHOULD be no greater than the requested Lifetime of the Registration Request. The minimum value SHOULD be large enough to account for the size of the messages, twice the round trip time for transmission to the home agent, and at least an additional 100 milliseconds to allow for processing the messages before responding. The round trip time for transmission to the home agent will be at least as large as the time required to transmit the messages at the link speed of the mobile node's current point of attachment. Some circuits add another 200 milliseconds of satellite delay in the total round trip time to the home agent. The minimum time between Registration Requests MUST NOT be less than 1 second. Each successive retransmission timeout period SHOULD be at least twice the previous period, as long as that is less than the maximum as specified above.

3.7. Foreign Agent Considerations

The foreign agent plays a mostly passive role in Mobile IP registration. It relays Registration Requests between mobile nodes and home agents, and, when it provides the care-of address, decapsulates datagrams for delivery to the mobile node. It SHOULD also send periodic Agent Advertisement messages to advertise its

presence as described in Section 2.3, if not detectable by link-layer means.

A foreign agent **MUST NOT** transmit a Registration Request except when relaying a Registration Request received from a mobile node, to the mobile node's home agent. A foreign agent **MUST NOT** transmit a Registration Reply except when relaying a Registration Reply received from a mobile node's home agent, or when replying to a Registration Request received from a mobile node in the case in which the foreign agent is denying service to the mobile node. In particular, a foreign agent **MUST NOT** generate a Registration Request or Reply because a mobile node's registration Lifetime has expired. A foreign agent also **MUST NOT** originate a Registration Request message that asks for deregistration of a mobile node; however, it **MUST** relay valid (de)Registration Requests originated by a mobile node.

3.7.1. Configuration and Registration Tables

Each foreign agent **MUST** be configured with a care-of address. In addition, for each pending or current registration the foreign agent **MUST** maintain a visitor list entry containing the following information obtained from the mobile node's Registration Request:

- the link-layer source address of the mobile node
- the IP Source Address (the mobile node's Home Address) or its co-located care-of address (see description of the 'R' bit in section 2.1.1)
- the IP Destination Address (as specified in 3.6.1.1)
- the UDP Source Port
- the Home Agent address
- the Identification field
- the requested registration Lifetime, and
- the remaining Lifetime of the pending or current registration.

If the mobile node's Home Address is zero in the Registration Request message, then the foreign agent **MUST** follow the procedures specified in RFC 2794 [6]. In particular, if the foreign agent cannot manage pending registration request records with such a zero Home Address for the mobile node, the foreign agent **MUST** return a Registration Reply with Code indicating `NONZERO_HOMEADDR_REQD` (see [6]).

The foreign agent **MAY** configure a maximum number of pending registrations that it is willing to maintain (typically 5). Additional registrations **SHOULD** then be rejected by the foreign agent with code 66. The foreign agent **MAY** delete any pending Registration Request after the request has been pending for more than 7 seconds; in this case, the foreign agent **SHOULD** reject the Request with code 78 (registration timeout).

As with any node on the Internet, a foreign agent MAY also share mobility security associations with any other nodes. When relaying a Registration Request from a mobile node to its home agent, if the foreign agent shares a mobility security association with the home agent, it MUST add a Foreign-Home Authentication Extension to the Request and MUST check the required Foreign-Home Authentication Extension in the Registration Reply from the home agent (Sections 3.3 and 3.4). Similarly, when receiving a Registration Request from a mobile node, if the foreign agent shares a mobility security association with the mobile node, it MUST check the required Mobile-Foreign Authentication Extension in the Request and MUST add a Mobile-Foreign Authentication Extension to the Registration Reply to the mobile node.

3.7.2. Receiving Registration Requests

If the foreign agent accepts a Registration Request from a mobile node, it checks to make sure that the indicated home agent address does not belong to any network interface of the foreign agent. If not, the foreign agent then MUST relay the Request to the indicated home agent. Otherwise, if the foreign agent denies the Request, it MUST send a Registration Reply to the mobile node with an appropriate denial Code, except in cases where the foreign agent would be required to send out more than one such denial per second to the same mobile node. The following sections describe this behavior in more detail.

If the foreign agent has configured one of its network interfaces with the IP address specified by the mobile node as its home agent address, the foreign agent MUST NOT forward the request again. If the foreign agent serves the mobile node as a home agent, the foreign agent follows the procedures specified in section 3.8.2. Otherwise, if the foreign agent does not serve the mobile node as a home agent, the foreign agent rejects the Registration Request with code 136 (unknown home agent address).

If a foreign agent receives a Registration Request from a mobile node in its visitor list, the existing visitor list entry for the mobile node SHOULD NOT be deleted or modified until the foreign agent receives a valid Registration Reply from the home agent with a Code indicating success. The foreign agent MUST record the new pending Request as a separate part of the existing visitor list entry for the mobile node. If the Registration Request requests deregistration, the existing visitor list entry for the mobile node SHOULD NOT be deleted until the foreign agent has received a successful Registration Reply. If the Registration Reply indicates that the

Request (for registration or deregistration) was denied by the home agent, the existing visitor list entry for the mobile node MUST NOT be modified as a result of receiving the Registration Reply.

3.7.2.1. Validity Checks

Registration Requests with an invalid, non-zero UDP checksum MUST be silently discarded. Requests with non-zero bits in reserved fields MUST be rejected with code 70 (poorly formed request). Requests with the 'D' bit set to 0, and specifying a care-of address not offered by the foreign agent, MUST be rejected with code 77 (invalid care-of address).

Also, the authentication in the Registration Request MUST be checked. If the foreign agent and the mobile node share a mobility security association, exactly one Mobile-Foreign Authentication Extension MUST be present in the Registration Request, and the foreign agent MUST check the Authenticator value in the Extension. If no Mobile-Foreign Authentication Extension is found, or if more than one Mobile-Foreign Authentication Extension is found, or if the Authenticator is invalid, the foreign agent MUST silently discard the Request and SHOULD log the event as a security exception. The foreign agent also SHOULD send a Registration Reply to the mobile node with Code 67.

3.7.2.2. Forwarding a Valid Request to the Home Agent

If the foreign agent accepts the mobile node's Registration Request, it MUST relay the Request to the mobile node's home agent as specified in the Home Agent field of the Registration Request. The foreign agent MUST NOT modify any of the fields beginning with the fixed portion of the Registration Request up through and including the Mobile-Home Authentication Extension or other authentication extension supplied by the mobile node as an authorization-enabling extension for the home agent. Otherwise, an authentication failure is very likely to occur at the home agent. In addition, the foreign agent proceeds as follows:

- It MUST process and remove any Extensions following the Mobile-Home Authentication Extension,
- It MAY append any of its own non-authentication Extensions of relevance to the home agent, if applicable, and
- It MUST append the Foreign-Home Authentication Extension, if the foreign agent shares a mobility security association with the home agent.

Specific fields within the IP header and the UDP header of the relayed Registration Request **MUST** be set as follows:

IP Source Address

The foreign agent's address on the interface from which the message will be sent.

IP Destination Address

Copied from the Home Agent field within the Registration Request.

UDP Source Port

<variable>

UDP Destination Port

434

After forwarding a valid Registration Request to the home agent, the foreign agent **MUST** begin timing the remaining lifetime of the pending registration based on the Lifetime in the Registration Request. If this lifetime expires before receiving a valid Registration Reply, the foreign agent **MUST** delete its visitor list entry for this pending registration.

3.7.2.3. Denying Invalid Requests

If the foreign agent denies the mobile node's Registration Request for any reason, it **SHOULD** send the mobile node a Registration Reply with a suitable denial Code. In such a case, the Home Address, Home Agent, and Identification fields within the Registration Reply are copied from the corresponding fields of the Registration Request.

If the Reserved field is nonzero, the foreign agent **MUST** deny the Request and **SHOULD** return a Registration Reply with status code 70 to the mobile node. If the Request is being denied because the requested Lifetime is too long, the foreign agent sets the Lifetime in the Reply to the maximum Lifetime value it is willing to accept in any Registration Request, and sets the Code field to 69. Otherwise, the Lifetime **SHOULD** be copied from the Lifetime field in the Request.

Specific fields within the IP header and the UDP header of the Registration Reply **MUST** be set as follows:

IP Source Address

Copied from the IP Destination Address of Registration Request, unless the "All Agents Multicast" address was used. In this case, the foreign agent's address (on the interface from which the message will be sent) **MUST** be used.

IP Destination Address

If the Registration Reply is generated by the Foreign Agent in order to reject a mobile node's Registration Request, and the Registration Request contains a Home Address which is not 0.0.0.0, then the IP Destination Address is copied from the Home Address field of the Registration Request. Otherwise, if the Registration Reply is received from the Home Agent, and contains a Home Address which is not 0.0.0.0, then the IP Destination Address is copied from the Home Address field of the Registration Reply. Otherwise, the IP Destination Address of the Registration Reply is set to be 255.255.255.255.

UDP Source Port

434

UDP Destination Port

Copied from the UDP Source Port of the Registration Request.

3.7.3. Receiving Registration Replies

The foreign agent updates its visitor list when it receives a valid Registration Reply from a home agent. It then relays the Registration Reply to the mobile node. The following sections describe this behavior in more detail.

If upon relaying a Registration Request to a home agent, the foreign agent receives an ICMP error message instead of a Registration Reply, then the foreign agent **SHOULD** send to the mobile node a Registration Reply with an appropriate "Home Agent Unreachable" failure Code (within the range 80-95, inclusive). See Section 3.7.2.3 for details on building the Registration Reply.

3.7.3.1. Validity Checks

Registration Replies with an invalid, non-zero UDP checksum **MUST** be silently discarded.

When a foreign agent receives a Registration Reply message, it **MUST** search its visitor list for a pending Registration Request with the same mobile node home address as indicated in the Reply. If no such pending Request is found, and if the Registration Reply does not correspond with any pending Registration Request with a zero mobile node home address (see section 3.7.1), the foreign agent **MUST** silently discard the Reply. The foreign agent **MUST** also silently discard the Reply if the low-order 32 bits of the Identification field in the Reply do not match those in the Request.

Also, the authentication in the Registration Reply **MUST** be checked. If the foreign agent and the home agent share a mobility security association, exactly one Foreign-Home Authentication Extension **MUST** be present in the Registration Reply, and the foreign agent **MUST** check the Authenticator value in the Extension. If no Foreign-Home Authentication Extension is found, or if more than one Foreign-Home Authentication Extension is found, or if the Authenticator is invalid, the foreign agent **MUST** silently discard the Reply and **SHOULD** log the event as a security exception. The foreign agent also **MUST** reject the mobile node's registration and **SHOULD** send a Registration Reply to the mobile node with Code 68.

3.7.3.2. Forwarding Replies to the Mobile Node

A Registration Reply which satisfies the validity checks of Section 3.8.2.1 is relayed to the mobile node. The foreign agent **MUST** also update its visitor list entry for the mobile node to reflect the results of the Registration Request, as indicated by the Code field in the Reply. If the Code indicates that the home agent has accepted the registration and the Lifetime field is nonzero, the foreign agent **SHOULD** set the Lifetime in the visitor list entry to the minimum of the following two values:

- the value specified in the Lifetime field of the Registration Reply, and
- the foreign agent's own maximum value for allowable registration lifetime.

If, instead, the Code indicates that the Lifetime field is zero, the foreign agent **MUST** delete its visitor list entry for the mobile node. Finally, if the Code indicates that the registration was denied by

the home agent, the foreign agent **MUST** delete its pending registration list entry, but not its visitor list entry, for the mobile node.

The foreign agent **MUST NOT** modify any of the fields beginning with the fixed portion of the Registration Reply up through and including the Mobile-Home Authentication Extension. Otherwise, an authentication failure is very likely to occur at the mobile node.

In addition, the foreign agent **SHOULD** perform the following additional procedures:

- It **MUST** process and remove any Extensions following the Mobile-Home Authentication Extension,
- It **MAY** append its own non-authentication Extensions of relevance to the mobile node, if applicable, and
- It **MUST** append the Mobile-Foreign Authentication Extension, if the foreign agent shares a mobility security association with the mobile node.

Specific fields within the IP header and the UDP header of the relayed Registration Reply are set according to the same rules specified in Section 3.7.2.3.

After forwarding a valid Registration Reply to the mobile node, the foreign agent **MUST** update its visitor list entry for this registration as follows. If the Registration Reply indicates that the registration was accepted by the home agent, the foreign agent resets its timer of the lifetime of the registration to the Lifetime granted in the Registration Reply; unlike the mobile node's timing of the registration lifetime as described in Section 3.6.2.2, the foreign agent considers this lifetime to begin when it forwards the Registration Reply message, ensuring that the foreign agent will not expire the registration before the mobile node does. On the other hand, if the Registration Reply indicates that the registration was rejected by the home agent, the foreign agent deletes its visitor list entry for this attempted registration.

3.8. Home Agent Considerations

Home agents play a reactive role in the registration process. The home agent receives Registration Requests from the mobile node (perhaps relayed by a foreign agent), updates its record of the mobility bindings for this mobile node, and issues a suitable Registration Reply in response to each.

A home agent **MUST NOT** transmit a Registration Reply except when replying to a Registration Request received from a mobile node. In particular, the home agent **MUST NOT** generate a Registration Reply to indicate that the Lifetime has expired.

3.8.1. Configuration and Registration Tables

Each home agent **MUST** be configured with an IP address and with the prefix size for the home network. The home agent **MUST** be configured with the mobility security association of each authorized mobile node that it is serving as a home agent.

When the home agent accepts a valid Registration Request from a mobile node that it serves as a home agent, the home agent **MUST** create or modify the entry for this mobile node in its mobility binding list containing:

- the mobile node's home address
- the mobile node's care-of address
- the Identification field from the Registration Reply
- the remaining Lifetime of the registration

The home agent **MAY** optionally offer the capability to dynamically associate a home address to a mobile node upon receiving a Registration Request from that mobile node. The method by which a home address is allocated to the mobile node is beyond the scope of this document, but see [6]. After the home agent makes the association of the home address to the mobile node, the home agent **MUST** put that home address into the Home Address field of the Registration Reply.

The home agent **MAY** also maintain mobility security associations with various foreign agents. When receiving a Registration Request from a foreign agent, if the home agent shares a mobility security association with the foreign agent, the home agent **MUST** check the Authenticator in the required Foreign-Home Authentication Extension in the message, based on this mobility security association. Similarly, when sending a Registration Reply to a foreign agent, if the home agent shares a mobility security association with the foreign agent, the home agent **MUST** include a Foreign-Home Authentication Extension in the message, based on this mobility security association.

3.8.2. Receiving Registration Requests

If the home agent accepts an incoming Registration Request, it **MUST** update its record of the the mobile node's mobility binding(s) and **SHOULD** send a Registration Reply with a suitable Code. Otherwise (the home agent denies the Request), it **SHOULD** send a Registration Reply with an appropriate Code specifying the reason the Request was denied. The following sections describe this behavior in more detail. If the home agent does not support broadcasts (see section 4.3), it **MUST** ignore the 'B' bit (as opposed to rejecting the Registration Request).

3.8.2.1. Validity Checks

Registration Requests with an invalid, non-zero UDP checksum **MUST** be silently discarded by the home agent.

The authentication in the Registration Request **MUST** be checked. This involves the following operations:

- a) The home agent **MUST** check for the presence of an authorization-enabling extension, and perform the indicated authentication. Exactly one authorization-enabling extension **MUST** be present in the Registration Request; and the home agent **MUST** either check the Authenticator value in the extension or verify that the authenticator value has been checked by another agent with which it has a security association. If no authorization-enabling extension is found, or if more than one authorization-enabling extension is found, or if the Authenticator is invalid, the home agent **MUST** reject the mobile node's registration and **SHOULD** send a Registration Reply to the mobile node with Code 131. The home agent **MUST** then discard the Request and **SHOULD** log the error as a security exception.
- b) The home agent **MUST** check that the registration Identification field is correct using the context selected by the SPI within the authorization-enabling extension. See Section 5.7 for a description of how this is performed. If incorrect, the home agent **MUST** reject the Request and **SHOULD** send a Registration Reply to the mobile node with Code 133, including an Identification field computed in accordance with the rules specified in Section 5.7. The home agent **MUST** do no further processing with such a Request, though it **SHOULD** log the error as a security exception.
- c) If the home agent shares a mobility security association with the foreign agent, the home agent **MUST** check for the presence of a valid Foreign-Home Authentication Extension. Exactly one

Foreign-Home Authentication Extension **MUST** be present in the Registration Request in this case, and the home agent **MUST** check the Authenticator value in the Extension. If no Foreign-Home Authentication Extension is found, or if more than one Foreign-Home Authentication Extension is found, or if the Authenticator is invalid, the home agent **MUST** reject the mobile node's registration and **SHOULD** send a Registration Reply to the mobile node with Code 132. The home agent **MUST** then discard the Request and **SHOULD** log the error as a security exception.

In addition to checking the authentication in the Registration Request, home agents **MUST** deny Registration Requests that are sent to the subnet-directed broadcast address of the home network (as opposed to being unicast to the home agent). The home agent **MUST** discard the Request and **SHOULD** return a Registration Reply with a Code of 136. In this case, the Registration Reply will contain the home agent's unicast address, so that the mobile node can re-issue the Registration Request with the correct home agent address.

Note that some routers change the IP destination address of a datagram from a subnet-directed broadcast address to 255.255.255.255 before injecting it into the destination subnet. In this case, home agents that attempt to pick up dynamic home agent discovery requests by binding a socket explicitly to the subnet-directed broadcast address will not see such packets. Home agent implementors should be prepared for both the subnet-directed broadcast address and 255.255.255.255 if they wish to support dynamic home agent discovery.

3.8.2.2. Accepting a Valid Request

If the Registration Request satisfies the validity checks in Section 3.8.2.1, and the home agent is able to accommodate the Request, the home agent **MUST** update its mobility binding list for the requesting mobile node and **MUST** return a Registration Reply to the mobile node.

In this case, the Reply Code will be either 0 if the home agent supports simultaneous mobility bindings, or 1 if it does not. See Section 3.8.3 for details on building the Registration Reply message.

The home agent updates its record of the mobile node's mobility bindings as follows, based on the fields in the Registration Request:

- If the Lifetime is zero and the Care-of Address equals the mobile node's home address, the home agent deletes all of the entries in the mobility binding list for the requesting mobile node. This is how a mobile node requests that its home agent cease providing mobility services.

- If the Lifetime is zero and the Care-of Address does not equal the mobile node's home address, the home agent deletes only the entry containing the specified Care-of Address from the mobility binding list for the requesting mobile node. Any other active entries containing other care-of addresses will remain active.
- If the Lifetime is nonzero, the home agent adds an entry containing the requested Care-of Address to the mobility binding list for the mobile node. If the 'S' bit is set and the home agent supports simultaneous mobility bindings, the previous mobility binding entries are retained. Otherwise, the home agent removes all previous entries in the mobility binding list for the mobile node.

In all cases, the home agent **MUST** send a Registration Reply to the source of the Registration Request, which might indeed be a different foreign agent than that whose care-of address is being (de)registered. If the home agent shares a mobility security association with the foreign agent whose care-of address is being deregistered, and that foreign agent is different from the one which relayed the Registration Request, the home agent **MAY** additionally send a Registration Reply to the foreign agent whose care-of address is being deregistered. The home agent **MUST NOT** send such a Reply if it does not share a mobility security association with the foreign agent. If no Reply is sent, the foreign agent's visitor list will expire naturally when the original Lifetime expires.

The home agent **MUST NOT** increase the Lifetime above that specified by the mobile node in the Registration Request. However, it is not an error for the mobile node to request a Lifetime longer than the home agent is willing to accept. In this case, the home agent simply reduces the Lifetime to a permissible value and returns this value in the Registration Reply. The Lifetime value in the Registration Reply informs the mobile node of the granted lifetime of the registration, indicating when it **SHOULD** re-register in order to maintain continued service. After the expiration of this registration lifetime, the home agent **MUST** delete its entry for this registration in its mobility binding list.

If the Registration Request duplicates an accepted current Registration Request, the new Lifetime **MUST NOT** extend beyond the Lifetime originally granted. A Registration Request is a duplicate if the home address, care-of address, and Identification fields all equal those of an accepted current registration.

In addition, if the home network implements ARP [36], and the Registration Request asks the home agent to create a mobility binding for a mobile node which previously had no binding (the mobile node was previously assumed to be at home), then the home agent **MUST** follow the procedures described in Section 4.6 with regard to ARP, proxy ARP, and gratuitous ARP. If the mobile node already had a previous mobility binding, the home agent **MUST** continue to follow the rules for proxy ARP described in Section 4.6.

3.8.2.3. Denying an Invalid Request

If the Registration Reply does not satisfy all of the validity checks in Section 3.8.2.1, or the home agent is unable to accommodate the Request, the home agent **SHOULD** return a Registration Reply to the mobile node with a Code that indicates the reason for the error. If a foreign agent was involved in relaying the Request, this allows the foreign agent to delete its pending visitor list entry. Also, this informs the mobile node of the reason for the error such that it may attempt to fix the error and issue another Request.

This section lists a number of reasons the home agent might reject a Request, and provides the Code value it should use in each instance. See Section 3.8.3 for additional details on building the Registration Reply message.

Many reasons for rejecting a registration are administrative in nature. For example, a home agent can limit the number of simultaneous registrations for a mobile node, by rejecting any registrations that would cause its limit to be exceeded, and returning a Registration Reply with error code 135. Similarly, a home agent may refuse to grant service to mobile nodes which have entered unauthorized service areas by returning a Registration Reply with a Code of 129.

Requests with non-zero bits in reserved fields **MUST** be rejected with code 134 (poorly formed request).

3.8.3. Sending Registration Replies

If the home agent accepts a Registration Request, it then **MUST** update its record of the mobile node's mobility binding(s) and **SHOULD** send a Registration Reply with a suitable Code. Otherwise (the home agent has denied the Request), it **SHOULD** send a Registration Reply with an appropriate Code specifying the reason the Request was denied. The following sections provide additional detail for the values the home agent **MUST** supply in the fields of Registration Reply messages.

3.8.3.1. IP/UDP Fields

This section provides the specific rules by which home agents pick values for the IP and UDP header fields of a Registration Reply.

IP Source Address

Copied from the IP Destination Address of Registration Request, unless a multicast or broadcast address was used. If the IP Destination Address of the Registration Request was a broadcast or multicast address, the IP Source Address of the Registration Reply MUST be set to the home agent's (unicast) IP address.

IP Destination Address

Copied from the IP Source Address of the Registration Request.

UDP Source Port

Copied from the UDP Destination Port of the Registration Request.

UDP Destination Port

Copied from the UDP Source Port of the Registration Request.

When sending a Registration Reply in response to a Registration Request that requested deregistration of the mobile node (the Lifetime is zero and the Care-of Address equals the mobile node's home address) and in which the IP Source Address was also set to the mobile node's home address (this is the normal method used by a mobile node to deregister when it returns to its home network), the IP Destination Address in the Registration Reply will be set to the mobile node's home address, as copied from the IP Source Address of the Request.

In this case, when transmitting the Registration Reply, the home agent MUST transmit the Reply directly onto the home network as if the mobile node were at home, bypassing any mobility binding list entry that may still exist at the home agent for the destination mobile node. In particular, for a mobile node returning home after being registered with a care-of address, if the mobile node's new Registration Request is not accepted by the home agent, the mobility binding list entry for the mobile node will still indicate that datagrams addressed to the mobile node should be tunneled to the mobile node's registered care-of address; when sending the Registration Reply indicating the rejection of this Request, this existing binding list entry MUST be ignored, and the home agent MUST transmit this Reply as if the mobile node were at home.

3.8.3.2. Registration Reply Fields

This section provides the specific rules by which home agents pick values for the fields within the fixed portion of a Registration Reply.

The Code field of the Registration Reply is chosen in accordance with the rules specified in the previous sections. When replying to an accepted registration, a home agent **SHOULD** respond with Code 1 if it does not support simultaneous registrations.

The Lifetime field **MUST** be copied from the corresponding field in the Registration Request, unless the requested value is greater than the maximum length of time the home agent is willing to provide the requested service. In such a case, the Lifetime **MUST** be set to the length of time that service will actually be provided by the home agent. This reduced Lifetime **SHOULD** be the maximum Lifetime allowed by the home agent (for this mobile node and care-of address).

If the Home Address field of the Registration Request is nonzero, it **MUST** be copied into the Home Address field of the Registration Reply message. Otherwise, if the Home Address field of the Registration Request is zero as specified in section 3.6, the home agent **SHOULD** arrange for the selection of a home address for the mobile node, and insert the selected address into the Home Address field of the Registration Reply message. See [6] for further relevant details in the case where mobile nodes identify themselves using an NAI instead of their IP home address.

If the Home Agent field in the Registration Request contains a unicast address of this home agent, then that field **MUST** be copied into the Home Agent field of the Registration Reply. Otherwise, the home agent **MUST** set the Home Agent field in the Registration Reply to its unicast address. In this latter case, the home agent **MUST** reject the registration with a suitable code (e.g., Code 136) to prevent the mobile node from possibly being simultaneously registered with two or more home agents.

3.8.3.3. Extensions

This section describes the ordering of any required and any optional Mobile IP Extensions that a home agent appends to a Registration Reply. The following ordering **MUST** be followed:

- a) The IP header, followed by the UDP header, followed by the fixed-length portion of the Registration Reply,

- b) If present, any non-authentication Extensions used by the mobile node (which may or may not also be used by the foreign agent),
- c) The Mobile-Home Authentication Extension,
- d) If present, any non-authentication Extensions used only by the foreign agent, and
- e) The Foreign-Home Authentication Extension, if present.

Note that items (a) and (c) **MUST** appear in every Registration Reply sent by the home agent. Items (b), (d), and (e) are optional. However, item (e) **MUST** be included when the home agent and the foreign agent share a mobility security association.

4. Routing Considerations

This section describes how mobile nodes, home agents, and (possibly) foreign agents cooperate to route datagrams to/from mobile nodes that are connected to a foreign network. The mobile node informs its home agent of its current location using the registration procedure described in Section 3. See the protocol overview in Section 1.7 for the relative locations of the mobile node's home address with respect to its home agent, and the mobile node itself with respect to any foreign agent with which it might attempt to register.

4.1. Encapsulation Types

Home agents and foreign agents **MUST** support tunneling datagrams using IP in IP encapsulation [32]. Any mobile node that uses a co-located care-of address **MUST** support receiving datagrams tunneled using IP in IP encapsulation. Minimal encapsulation [34] and GRE encapsulation [16] are alternate encapsulation methods which **MAY** optionally be supported by mobility agents and mobile nodes. The use of these alternative forms of encapsulation, when requested by the mobile node, is otherwise at the discretion of the home agent.

4.2. Unicast Datagram Routing

4.2.1. Mobile Node Considerations

When connected to its home network, a mobile node operates without the support of mobility services. That is, it operates in the same way as any other (fixed) host or router. The method by which a mobile node selects a default router when connected to its home

network, or when away from home and using a co-located care-of address, is outside the scope of this document. ICMP Router Advertisement [10] is one such method.

When registered on a foreign network, the mobile node chooses a default router by the following rules:

- If the mobile node is registered using a foreign agent care-of address, it MAY use its foreign agent as a first-hop router. The foreign agent's MAC address can be learned from Agent Advertisement. Otherwise, the mobile node MUST choose its default router from among the Router Addresses advertised in the ICMP Router Advertisement portion of that Agent Advertisement message.
- If the mobile node is registered directly with its home agent using a co-located care-of address, then the mobile node SHOULD choose its default router from among those advertised in any ICMP Router Advertisement message that it receives for which its externally obtained care-of address and the Router Address match under the network prefix. If the mobile node's externally obtained care-of address matches the IP source address of the Agent Advertisement under the network prefix, the mobile node MAY also consider that IP source address as another possible choice for the IP address of a default router. The network prefix MAY be obtained from the Prefix-Lengths Extension in the Router Advertisement, if present. The prefix MAY also be obtained through other mechanisms beyond the scope of this document.

While they are away from the home network, mobile nodes MUST NOT broadcast ARP packets to find the MAC address of another Internet node. Thus, the (possibly empty) list of Router Addresses from the ICMP Router Advertisement portion of the message is not useful for selecting a default router, unless the mobile node has some means not involving broadcast ARP and not specified within this document for obtaining the MAC address of one of the routers in the list. Similarly, in the absence of unspecified mechanisms for obtaining MAC addresses on foreign networks, the mobile node MUST ignore redirects to other routers on foreign networks.

4.2.2. Foreign Agent Considerations

Upon receipt of an encapsulated datagram sent to its advertised care-of address, a foreign agent MUST compare the inner destination address to those entries in its visitor list. When the destination does not match the address of any mobile node currently in the visitor list, the foreign agent MUST NOT forward the datagram without

modifications to the original IP header, because otherwise a routing loop is likely to result. The datagram SHOULD be silently discarded. ICMP Destination Unreachable MUST NOT be sent when a foreign agent is unable to forward an incoming tunneled datagram. Otherwise, the foreign agent forwards the decapsulated datagram to the mobile node.

The foreign agent MUST NOT advertise to other routers in its routing domain, nor to any other mobile node, the presence of a mobile router (Section 4.5) or mobile node in its visitor list.

The foreign agent MUST route datagrams it receives from registered mobile nodes. At a minimum, this means that the foreign agent must verify the IP Header Checksum, decrement the IP Time To Live, recompute the IP Header Checksum, and forward such datagrams to a default router.

A foreign agent MUST NOT use broadcast ARP for a mobile node's MAC address on a foreign network. It may obtain the MAC address by copying the information from an Agent Solicitation or a Registration Request transmitted from a mobile node. A foreign agent's ARP cache for the mobile node's IP address MUST NOT be allowed to expire before the mobile node's visitor list entry expires, unless the foreign agent has some way other than broadcast ARP to refresh its MAC address associated with the mobile node's IP address.

Each foreign agent SHOULD support the mandatory features for reverse tunneling [27].

4.2.3. Home Agent Considerations

The home agent MUST be able to intercept any datagrams on the home network addressed to the mobile node while the mobile node is registered away from home. Proxy and gratuitous ARP MAY be used in enabling this interception, as specified in Section 4.6.

The home agent must examine the IP Destination Address of all arriving datagrams to see if it is equal to the home address of any of its mobile nodes registered away from home. If so, the home agent tunnels the datagram to the mobile node's currently registered care-of address or addresses. If the home agent supports the optional capability of multiple simultaneous mobility bindings, it tunnels a copy to each care-of address in the mobile node's mobility binding list. If the mobile node has no current mobility bindings, the home agent MUST NOT attempt to intercept datagrams destined for the mobile node, and thus will not in general receive such datagrams. However, if the home agent is also a router handling common IP traffic, it is possible that it will receive such datagrams for forwarding onto the

home network. In this case, the home agent **MUST** assume the mobile node is at home and simply forward the datagram directly onto the home network.

For multihomed home agents, the source address in the outer IP header of the encapsulated datagram **MUST** be the address sent to the mobile node in the home agent field of the registration reply. That is, the home agent cannot use the the address of some other network interface as the source address.

See Section 4.1 regarding methods of encapsulation that may be used for tunneling. Nodes implementing tunneling **SHOULD** also implement the "tunnel soft state" mechanism [32], which allows ICMP error messages returned from the tunnel to correctly be reflected back to the original senders of the tunneled datagrams.

Home agents **MUST** decapsulate packets addressed to themselves, sent by a mobile node for the purpose of maintaining location privacy, as described in Section 5.5. This feature is also required for support of reverse tunneling [27].

If the Lifetime for a given mobility binding expires before the home agent has received another valid Registration Request for that mobile node, then that binding is deleted from the mobility binding list. The home agent **MUST NOT** send any Registration Reply message simply because the mobile node's binding has expired. The entry in the visitor list of the mobile node's current foreign agent will expire naturally, probably at the same time as the binding expired at the home agent. When a mobility binding's lifetime expires, the home agent **MUST** delete the binding, but it **MUST** retain any other (non-expired) simultaneous mobility bindings that it holds for the mobile node.

When a home agent receives a datagram, intercepted for one of its mobile nodes registered away from home, the home agent **MUST** examine the datagram to check if it is already encapsulated. If so, special rules apply in the forwarding of that datagram to the mobile node:

- If the inner (encapsulated) Destination Address is the same as the outer Destination Address (the mobile node), then the home agent **MUST** also examine the outer Source Address of the encapsulated datagram (the source address of the tunnel). If this outer Source Address is the same as the mobile node's current care-of address, the home agent **MUST** silently discard that datagram in order to prevent a likely routing loop. If, instead, the outer Source Address is **NOT** the same as the mobile node's current care-of address, then the home agent **SHOULD** forward the datagram to the mobile node. In order to forward

the datagram in this case, the home agent MAY simply alter the outer Destination Address to the care-of address, rather than re-encapsulating the datagram.

- Otherwise (the inner Destination Address is NOT the same as the outer Destination Address), the home agent SHOULD encapsulate the datagram again (nested encapsulation), with the new outer Destination Address set equal to the mobile node's care-of address. That is, the home agent forwards the entire datagram to the mobile node in the same way as any other datagram (encapsulated already or not).

4.3. Broadcast Datagrams

When a home agent receives a broadcast datagram, it MUST NOT forward the datagram to any mobile nodes in its mobility binding list other than those that have requested forwarding of broadcast datagrams. A mobile node MAY request forwarding of broadcast datagrams by setting the 'B' bit in its Registration Request message (Section 3.3). For each such registered mobile node, the home agent SHOULD forward received broadcast datagrams to the mobile node, although it is a matter of configuration at the home agent as to which specific categories of broadcast datagrams will be forwarded to such mobile nodes.

If the 'D' bit was set in the mobile node's Registration Request message, indicating that the mobile node is using a co-located care-of address, the home agent simply tunnels appropriate broadcast IP datagrams to the mobile node's care-of address. Otherwise (the 'D' bit was NOT set), the home agent first encapsulates the broadcast datagram in a unicast datagram addressed to the mobile node's home address, and then tunnels this encapsulated datagram to the foreign agent. This extra level of encapsulation is required so that the foreign agent can determine which mobile node should receive the datagram after it is decapsulated. When received by the foreign agent, the unicast encapsulated datagram is detunneled and delivered to the mobile node in the same way as any other datagram. In either case, the mobile node must decapsulate the datagram it receives in order to recover the original broadcast datagram.

4.4. Multicast Datagram Routing

As mentioned previously, a mobile node that is connected to its home network functions in the same way as any other (fixed) host or router. Thus, when it is at home, a mobile node functions identically to other multicast senders and receivers. This section therefore describes the behavior of a mobile node that is visiting a foreign network.

In order to receive multicasts, a mobile node **MUST** join the multicast group in one of two ways. First, a mobile node **MAY** join the group via a (local) multicast router on the visited subnet. This option assumes that there is a multicast router present on the visited subnet. If the mobile node is using a co-located care-of address, it **SHOULD** use this address as the source IP address of its IGMP [11] messages. Otherwise, it **MAY** use its home address.

Alternatively, a mobile node which wishes to receive multicasts **MAY** join groups via a bi-directional tunnel to its home agent, assuming that its home agent is a multicast router. The mobile node tunnels IGMP messages to its home agent and the home agent forwards multicast datagrams down the tunnel to the mobile node. For packets tunneled to the home agent, the source address in the IP header **SHOULD** be the mobile node's home address.

The rules for multicast datagram delivery to mobile nodes in this case are identical to those for broadcast datagrams (Section 4.3). Namely, if the mobile node is using a co-located care-of address (the 'D' bit was set in the mobile node's Registration Request), then the home agent **SHOULD** tunnel the datagram to this care-of address; otherwise, the home agent **MUST** first encapsulate the datagram in a unicast datagram addressed to the mobile node's home address and then **MUST** tunnel the resulting datagram (nested tunneling) to the mobile node's care-of address. For this reason, the mobile node **MUST** be capable of decapsulating packets sent to its home address in order to receive multicast datagrams using this method.

A mobile node that wishes to send datagrams to a multicast group also has two options: (1) send directly on the visited network; or (2) send via a tunnel to its home agent. Because multicast routing in general depends upon the IP source address, a mobile node which sends multicast datagrams directly on the visited network **MUST** use a co-located care-of address as the IP source address. Similarly, a mobile node which tunnels a multicast datagram to its home agent **MUST** use its home address as the IP source address of both the (inner) multicast datagram and the (outer) encapsulating datagram. This second option assumes that the home agent is a multicast router.

4.5. Mobile Routers

A mobile node can be a router that is responsible for the mobility of one or more entire networks moving together, perhaps on an airplane, a ship, a train, an automobile, a bicycle, or a kayak. The nodes connected to a network served by the mobile router may themselves be fixed nodes or mobile nodes or routers. In this document, such networks are called "mobile networks".

A mobile router MAY act as a foreign agent and provide a foreign agent care-of address to mobile nodes connected to the mobile network. Typical routing to a mobile node via a mobile router in this case is illustrated by the following example:

- a) A laptop computer is disconnected from its home network and later attached to a network port in the seat back of an aircraft. The laptop computer uses Mobile IP to register on this foreign network, using a foreign agent care-of address discovered through an Agent Advertisement from the aircraft's foreign agent.
- b) The aircraft network is itself mobile. Suppose the node serving as the foreign agent on the aircraft also serves as the default router that connects the aircraft network to the rest of the Internet. When the aircraft is at home, this router is attached to some fixed network at the airline's headquarters, which is the router's home network. While the aircraft is in flight, this router registers from time to time over its radio link with a series of foreign agents below it on the ground. This router's home agent is a node on the fixed network at the airline's headquarters.
- c) Some correspondent node sends a datagram to the laptop computer, addressing the datagram to the laptop's home address. This datagram is initially routed to the laptop's home network.
- d) The laptop's home agent intercepts the datagram on the home network and tunnels it to the laptop's care-of address, which in this example is an address of the node serving as router and foreign agent on the aircraft. Normal IP routing will route the datagram to the fixed network at the airline's headquarters.
- e) The aircraft router and foreign agent's home agent there intercepts the datagram and tunnels it to its current care-of address, which in this example is some foreign agent on the ground below the aircraft. The original datagram from the correspondent node has now been encapsulated twice: once by the laptop's home agent and again by the aircraft's home agent.
- f) The foreign agent on the ground decapsulates the datagram, yielding a datagram still encapsulated by the laptop's home agent, with a destination address of the laptop's care-of address. The ground foreign agent sends the resulting datagram over its radio link to the aircraft.

- g) The foreign agent on the aircraft decapsulates the datagram, yielding the original datagram from the correspondent node, with a destination address of the laptop's home address. The aircraft foreign agent delivers the datagram over the aircraft network to the laptop's link-layer address.

This example illustrated the case in which a mobile node is attached to a mobile network. That is, the mobile node is mobile with respect to the network, which itself is also mobile (here with respect to the ground). If, instead, the node is fixed with respect to the mobile network (the mobile network is the fixed node's home network), then either of two methods may be used to cause datagrams from correspondent nodes to be routed to the fixed node.

A home agent MAY be configured to have a permanent registration for the fixed node, that indicates the mobile router's address as the fixed host's care-of address. The mobile router's home agent will usually be used for this purpose. The home agent is then responsible for advertising connectivity using normal routing protocols to the fixed node. Any datagrams sent to the fixed node will thus use nested tunneling as described above.

Alternatively, the mobile router MAY advertise connectivity to the entire mobile network using normal IP routing protocols through a bi-directional tunnel to its own home agent. This method avoids the need for nested tunneling of datagrams.

4.6. ARP, Proxy ARP, and Gratuitous ARP

The use of ARP [36] requires special rules for correct operation when wireless or mobile nodes are involved. The requirements specified in this section apply to all home networks in which ARP is used for address resolution.

In addition to the normal use of ARP for resolving a target node's link-layer address from its IP address, this document distinguishes two special uses of ARP:

- A Proxy ARP [39] is an ARP Reply sent by one node on behalf of another node which is either unable or unwilling to answer its own ARP Requests. The sender of a Proxy ARP reverses the Sender and Target Protocol Address fields as described in [36], but supplies some configured link-layer address (generally, its own) in the Sender Hardware Address field. The node receiving the Reply will then associate this link-layer address with the IP address of the original target node, causing it to transmit future datagrams for this target node to the node with that link-layer address.

- A Gratuitous ARP [45] is an ARP packet sent by a node in order to spontaneously cause other nodes to update an entry in their ARP cache. A gratuitous ARP MAY use either an ARP Request or an ARP Reply packet. In either case, the ARP Sender Protocol Address and ARP Target Protocol Address are both set to the IP address of the cache entry to be updated, and the ARP Sender Hardware Address is set to the link-layer address to which this cache entry should be updated. When using an ARP Reply packet, the Target Hardware Address is also set to the link-layer address to which this cache entry should be updated (this field is not used in an ARP Request packet).

In either case, for a gratuitous ARP, the ARP packet MUST be transmitted as a local broadcast packet on the local link. As specified in [36], any node receiving any ARP packet (Request or Reply) MUST update its local ARP cache with the Sender Protocol and Hardware Addresses in the ARP packet, if the receiving node has an entry for that IP address already in its ARP cache. This requirement in the ARP protocol applies even for ARP Request packets, and for ARP Reply packets that do not match any ARP Request transmitted by the receiving node [36].

While a mobile node is registered on a foreign network, its home agent uses proxy ARP [39] to reply to ARP Requests it receives that seek the mobile node's link-layer address. When receiving an ARP Request, the home agent MUST examine the target IP address of the Request, and if this IP address matches the home address of any mobile node for which it has a registered mobility binding, the home agent MUST transmit an ARP Reply on behalf of the mobile node. After exchanging the sender and target addresses in the packet [39], the home agent MUST set the sender link-layer address in the packet to the link-layer address of its own interface over which the Reply will be sent.

When a mobile node leaves its home network and registers a binding on a foreign network, its home agent uses gratuitous ARP to update the ARP caches of nodes on the home network. This causes such nodes to associate the link-layer address of the home agent with the mobile node's home (IP) address. When registering a binding for a mobile node for which the home agent previously had no binding (the mobile node was assumed to be at home), the home agent MUST transmit a gratuitous ARP on behalf of the mobile node. This gratuitous ARP packet MUST be transmitted as a broadcast packet on the link on which the mobile node's home address is located. Since broadcasts on the local link (such as Ethernet) are typically not guaranteed to be reliable, the gratuitous ARP packet SHOULD be retransmitted a small number of times to increase its reliability.

When a mobile node returns to its home network, the mobile node and its home agent use gratuitous ARP to cause all nodes on the mobile node's home network to update their ARP caches to once again associate the mobile node's own link-layer address with the mobile node's home (IP) address. Before transmitting the (de)Registration Request message to its home agent, the mobile node **MUST** transmit this gratuitous ARP on its home network as a local broadcast on this link. The gratuitous ARP packet **SHOULD** be retransmitted a small number of times to increase its reliability, but these retransmissions **SHOULD** proceed in parallel with the transmission and processing of its (de)Registration Request.

When the mobile node's home agent receives and accepts this (de)Registration Request, the home agent **MUST** also transmit a gratuitous ARP on the mobile node's home network. This gratuitous ARP also is used to associate the mobile node's home address with the mobile node's own link-layer address. A gratuitous ARP is transmitted by both the mobile node and its home agent, since in the case of wireless network interfaces, the area within transmission range of the mobile node will likely differ from that within range of its home agent. The ARP packet from the home agent **MUST** be transmitted as a local broadcast on the mobile node's home link, and **SHOULD** be retransmitted a small number of times to increase its reliability; these retransmissions, however, **SHOULD** proceed in parallel with the transmission and processing of its (de)Registration Reply.

While the mobile node is away from home, it **MUST NOT** transmit any broadcast ARP Request or ARP Reply messages. Finally, while the mobile node is away from home, it **MUST NOT** reply to ARP Requests in which the target IP address is its own home address, unless the ARP Request is unicast by a foreign agent with which the mobile node is attempting to register or a foreign agent with which the mobile node has an unexpired registration. In the latter case, the mobile node **MUST** use a unicast ARP Reply to respond to the foreign agent. Note that if the mobile node is using a co-located care-of address and receives an ARP Request in which the target IP address is this care-of address, then the mobile node **SHOULD** reply to this ARP Request. Note also that, when transmitting a Registration Request on a foreign network, a mobile node may discover the link-layer address of a foreign agent by storing the address as it is received from the Agent Advertisement from that foreign agent, but not by transmitting a broadcast ARP Request message.

The specific order in which each of the above requirements for the use of ARP, proxy ARP, and gratuitous ARP are applied, relative to the transmission and processing of the mobile node's Registration Request and Registration Reply messages when leaving home or returning home, are important to the correct operation of the protocol.

To summarize the above requirements, when a mobile node leaves its home network, the following steps, in this order, **MUST** be performed:

- The mobile node decides to register away from home, perhaps because it has received an Agent Advertisement from a foreign agent and has not recently received one from its home agent.
- Before transmitting the Registration Request, the mobile node disables its own future processing of any ARP Requests it may subsequently receive requesting the link-layer address corresponding to its home address, except insofar as necessary to communicate with foreign agents on visited networks.
- The mobile node transmits its Registration Request.
- When the mobile node's home agent receives and accepts the Registration Request, it performs a gratuitous ARP on behalf of the mobile node, and begins using proxy ARP to reply to ARP Requests that it receives requesting the mobile node's link-layer address. In the gratuitous ARP, the ARP Sender Hardware Address is set to the link-layer address of the home agent. If, instead, the home agent rejects the Registration Request, no ARP processing (gratuitous nor proxy) is performed by the home agent.

When a mobile node later returns to its home network, the following steps, in this order, **MUST** be performed:

- The mobile node decides to register at home, perhaps because it has received an Agent Advertisement from its home agent.
- Before transmitting the Registration Request, the mobile node re-enables its own future processing of any ARP Requests it may subsequently receive requesting its link-layer address.
- The mobile node performs a gratuitous ARP for itself. In this gratuitous ARP, the ARP Sender Hardware Address is set to the link-layer address of the mobile node.
- The mobile node transmits its Registration Request.

- When the mobile node's home agent receives and accepts the Registration Request, it stops using proxy ARP to reply to ARP Requests that it receives requesting the mobile node's link-layer address, and then performs a gratuitous ARP on behalf of the mobile node. In this gratuitous ARP, the ARP Sender Hardware Address is set to the link-layer address of the mobile node. If, instead, the home agent rejects the Registration Request, the home agent **MUST NOT** make any change to the way it performs ARP processing (gratuitous nor proxy) for the mobile node. In this latter case, the home agent should operate as if the mobile node has not returned home, and continue to perform proxy ARP on behalf of the mobile node.

5. Security Considerations

The mobile computing environment is potentially very different from the ordinary computing environment. In many cases, mobile computers will be connected to the network via wireless links. Such links are particularly vulnerable to passive eavesdropping, active replay attacks, and other active attacks.

5.1. Message Authentication Codes

Home agents and mobile nodes **MUST** be able to perform authentication. The default algorithm is HMAC-MD5 [23], with a key size of 128 bits. The foreign agent **MUST** also support authentication using HMAC-MD5 and key sizes of 128 bits or greater, with manual key distribution. Keys with arbitrary binary values **MUST** be supported.

The "prefix+suffix" use of MD5 to protect data and a shared secret is considered vulnerable to attack by the cryptographic community. Where backward compatibility with existing Mobile IP implementations that use this mode is needed, new implementations **SHOULD** include keyed MD5 [41] as one of the additional authentication algorithms for use when producing and verifying the authentication data that is supplied with Mobile IP registration messages, for instance in the extensions specified in sections 3.5.2, 3.5.3, and 3.5.4.

More authentication algorithms, algorithm modes, key distribution methods, and key sizes **MAY** also be supported for all of these extensions.

5.2. Areas of Security Concern in this Protocol

The registration protocol described in this document will result in a mobile node's traffic being tunneled to its care-of address. This tunneling feature could be a significant vulnerability if the registration were not authenticated. Such remote redirection, for

instance as performed by the mobile registration protocol, is widely understood to be a security problem in the current Internet if not authenticated [2]. Moreover, the Address Resolution Protocol (ARP) is not authenticated, and can potentially be used to steal another host's traffic. The use of "Gratuitous ARP" (Section 4.6) brings with it all of the risks associated with the use of ARP.

5.3. Key Management

This specification requires a strong authentication mechanism (keyed MD5) which precludes many potential attacks based on the Mobile IP registration protocol. However, because key distribution is difficult in the absence of a network key management protocol, messages with the foreign agent are not all required to be authenticated. In a commercial environment it might be important to authenticate all messages between the foreign agent and the home agent, so that billing is possible, and service providers do not provide service to users that are not legitimate customers of that service provider.

5.4. Picking Good Random Numbers

The strength of any authentication mechanism depends on several factors, including the innate strength of the authentication algorithm, the secrecy of the key used, the strength of the key used, and the quality of the particular implementation. This specification requires implementation of keyed MD5 for authentication, but does not preclude the use of other authentication algorithms and modes. For keyed MD5 authentication to be useful, the 128-bit key must be both secret (that is, known only to authorized parties) and pseudo-random. If nonces are used in connection with replay protection, they must also be selected carefully. Eastlake, et al. [14] provides more information on generating pseudo-random numbers.

5.5. Privacy

Users who have sensitive data that they do not wish others to see should use mechanisms outside the scope of this document (such as encryption) to provide appropriate protection. Users concerned about traffic analysis should consider appropriate use of link encryption. If absolute location privacy is desired, the mobile node can create a tunnel to its home agent. Then, datagrams destined for correspondent nodes will appear to emanate from the home network, and it may be more difficult to pinpoint the location of the mobile node. Such mechanisms are all beyond the scope of this document.

5.6. Ingress Filtering

Many routers implement security policies such as "ingress filtering" [15] that do not allow forwarding of packets that have a Source Address which appears topologically incorrect. In environments where this is a problem, mobile nodes may use reverse tunneling [27] with the foreign agent supplied care-of address as the Source Address. Reverse tunneled packets will be able to pass normally through such routers, while ingress filtering rules will still be able to locate the true topological source of the packet in the same way as packets from non-mobile nodes.

5.7. Replay Protection for Registration Requests

The Identification field is used to let the home agent verify that a registration message has been freshly generated by the mobile node, not replayed by an attacker from some previous registration. Two methods are described in this section: timestamps (mandatory) and "nonces" (optional). All mobile nodes and home agents **MUST** implement timestamp-based replay protection. These nodes **MAY** also implement nonce-based replay protection (but see Appendix A).

The style of replay protection in effect between a mobile node and its home agent is part of the mobile security association. A mobile node and its home agent **MUST** agree on which method of replay protection will be used. The interpretation of the Identification field depends on the method of replay protection as described in the subsequent subsections.

Whatever method is used, the low-order 32 bits of the Identification **MUST** be copied unchanged from the Registration Request to the Reply. The foreign agent uses those bits (and the mobile node's home address) to match Registration Requests with corresponding replies. The mobile node **MUST** verify that the low-order 32 bits of any Registration Reply are identical to the bits it sent in the Registration Request.

The Identification in a new Registration Request **MUST NOT** be the same as in an immediately preceding Request, and **SHOULD NOT** repeat while the same security context is being used between the mobile node and the home agent. Retransmission as in Section 3.6.3 is allowed.

5.7.1. Replay Protection using Timestamps

The basic principle of timestamp replay protection is that the node generating a message inserts the current time of day, and the node receiving the message checks that this timestamp is sufficiently close to its own time of day. Unless specified differently in the

security association between the nodes, a default value of 7 seconds MAY be used to limit the time difference. This value SHOULD be greater than 3 seconds. Obviously the two nodes must have adequately synchronized time-of-day clocks. As with any messages, time synchronization messages may be protected against tampering by an authentication mechanism determined by the security context between the two nodes.

If timestamps are used, the mobile node MUST set the Identification field to a 64-bit value formatted as specified by the Network Time Protocol [26]. The low-order 32 bits of the NTP format represent fractional seconds, and those bits which are not available from a time source SHOULD be generated from a good source of randomness. Note, however, that when using timestamps, the 64-bit Identification used in a Registration Request from the mobile node MUST be greater than that used in any previous Registration Request, as the home agent uses this field also as a sequence number. Without such a sequence number, it would be possible for a delayed duplicate of an earlier Registration Request to arrive at the home agent (within the clock synchronization required by the home agent), and thus be applied out of order, mistakenly altering the mobile node's current registered care-of address.

Upon receipt of a Registration Request with an authorization-enabling extension, the home agent MUST check the Identification field for validity. In order to be valid, the timestamp contained in the Identification field MUST be close enough to the home agent's time of day clock and the timestamp MUST be greater than all previously accepted timestamps for the requesting mobile node. Time tolerances and resynchronization details are specific to a particular mobility security association.

If the timestamp is valid, the home agent copies the entire Identification field into the Registration Reply it returns the Reply to the mobile node. If the timestamp is not valid, the home agent copies only the low-order 32 bits into the Registration Reply, and supplies the high-order 32 bits from its own time of day. In this latter case, the home agent MUST reject the registration by returning Code 133 (identification mismatch) in the Registration Reply.

As described in Section 3.6.2.1, the mobile node MUST verify that the low-order 32 bits of the Identification in the Registration Reply are identical to those in the rejected registration attempt, before using the high-order bits for clock resynchronization.

5.7.2. Replay Protection using Nonces

The basic principle of nonce replay protection is that node A includes a new random number in every message to node B, and checks that node B returns that same number in its next message to node A. Both messages use an authentication code to protect against alteration by an attacker. At the same time node B can send its own nonces in all messages to node A (to be echoed by node A), so that it too can verify that it is receiving fresh messages.

The home agent may be expected to have resources for computing pseudo-random numbers useful as nonces [14]. It inserts a new nonce as the high-order 32 bits of the identification field of every Registration Reply. The home agent copies the low-order 32 bits of the Identification from the Registration Request message into the low-order 32 bits of the Identification in the Registration Reply. When the mobile node receives an authenticated Registration Reply from the home agent, it saves the high-order 32 bits of the identification for use as the high-order 32 bits of its next Registration Request.

The mobile node is responsible for generating the low-order 32 bits of the Identification in each Registration Request. Ideally it should generate its own random nonces. However it may use any expedient method, including duplication of the random value sent by the home agent. The method chosen is of concern only to the mobile node, because it is the node that checks for valid values in the Registration Reply. The high-order and low-order 32 bits of the identification chosen SHOULD both differ from their previous values. The home agent uses a new high-order value and the mobile node uses a new low-order value for each registration message. The foreign agent uses the low-order value (and the mobile host's home address) to correctly match registration replies with pending Requests (Section 3.7.1).

If a registration message is rejected because of an invalid nonce, the Reply always provides the mobile node with a new nonce to be used in the next registration. Thus the nonce protocol is self-synchronizing.

6. IANA Considerations

Mobile IP specifies several new number spaces for values to be used in various message fields. These number spaces include the following:

- Mobile IP message types sent to UDP port 434, as defined in section 1.8.

- types of extensions to Registration Request and Registration Reply messages (see sections 3.3 and 3.4, and also consult [27, 29, 6, 7, 12])
- values for the Code in the Registration Reply message (see section 3.4, and also consult [27, 29, 6, 7, 12])
- Mobile IP defines so-called Agent Solicitation and Agent Advertisement messages. These messages are in fact Router Discovery messages [10] augmented with mobile-IP specific extensions. Thus, they do not define a new name space, but do define additional Router Discovery extensions as described below in Section 6.2. Also see Section 2.1 and consult [7, 12].

There are additional Mobile IP numbering spaces specified in [7].

Information about assignment of mobile-ip numbers derived from specifications external to this document is given by IANA at <http://www.iana.org/numbers.html>. From that URL, follow the hyperlinks to [M] within the "Directory of General Assigned Numbers", and subsequently to the specific section for "Mobile IP Numbers".

6.1. Mobile IP Message Types

Mobile IP messages are defined to be those that are sent to a message recipient at port 434 (UDP or TCP). The number space for Mobile IP messages is specified in Section 1.8. Approval of new extension numbers is subject to Expert Review, and a specification is required [30]. The currently standardized message types have the following numbers, and are specified in the indicated sections.

Type	Name	Section
1	Registration Request	3.3
3	Registration Reply	3.4

6.2. Extensions to RFC 1256 Router Advertisement

RFC 1256 defines two ICMP message types, Router Advertisement and Router Solicitation. Mobile IP defines a number space for extensions to Router Advertisement, which could be used by protocols other than Mobile IP. The extension types currently standardized for use with Mobile IP have the following numbers.

Type	Name	Reference
0	One-byte Padding	2.1.3
16	Mobility Agent Advertisement	2.1.1
19	Prefix-Lengths	2.1.2

Approval of new extension numbers for use with Mobile IP is subject to Expert Review, and a specification is required [30].

6.3. Extensions to Mobile IP Registration Messages

The Mobile IP messages, specified within this document, and listed in sections 1.8 and 6.1, may have extensions. Mobile IP message extensions all share the same number space, even if they are to be applied to different Mobile IP messages. The number space for Mobile IP message extensions is specified within this document. Approval of new extension numbers is subject to Expert Review, and a specification is required [30].

Type	Name	Reference
0	One-byte Padding	
32	Mobile-Home Authentication	3.5.2
33	Mobile-Foreign Authentication	3.5.3
34	Foreign-Home Authentication	3.5.4

6.4. Code Values for Mobile IP Registration Reply Messages

The Mobile IP Registration Reply message, specified in section 3.4, has a Code field. The number space for the Code field values is also specified in Section 3.4. The Code number space is structured according to whether the registration was successful, or whether the foreign agent denied the registration request, or lastly whether the home agent denied the registration request, as follows:

0-8	Success Codes
9-63	No allocation guidelines currently exist
64-127	Error Codes from the Foreign Agent
128-192	Error Codes from the Home Agent
193-255	No allocation guidelines currently exist

Approval of new Code values requires Expert Review [30].

7. Acknowledgments

Special thanks to Steve Deering (Xerox PARC), along with Dan Duchamp and John Ioannidis (JI) (Columbia University), for forming the working group, chairing it, and putting so much effort into its early development. Columbia's early Mobile IP work can be found in [18, 19, 17].

Thanks also to Kannan Alaggapan, Greg Minshall, Tony Li, Jim Solomon, Erik Nordmark, Basavaraj Patil, and Phil Roberts for their contributions to the group while performing the duties of chairperson, as well as for their many useful comments.

Thanks to the active members of the Mobile IP Working Group, particularly those who contributed text, including (in alphabetical order)

- Ran Atkinson (Naval Research Lab),
- Samita Chakrabarti (Sun Microsystems)
- Ken Imboden (Candlestick Networks, Inc.)
- Dave Johnson (Carnegie Mellon University),
- Frank Kastenholz (FTP Software),
- Anders Klemets (KTH),
- Chip Maguire (KTH),
- Alison Mankin (ISI)
- Andrew Myles (Macquarie University),
- Thomas Narten (IBM)
- Al Quirt (Bell Northern Research),
- Yakov Rekhter (IBM), and
- Fumio Teraoka (Sony).
- Alper Yegin (NTT DoCoMo)

Thanks to Charlie Kunzinger and to Bill Simpson, the editors who produced the first drafts for of this document, reflecting the discussions of the Working Group. Much of the new text in the later revisions preceding RFC 2002 is due to Jim Solomon and Dave Johnson.

Thanks to Greg Minshall (Novell), Phil Karn (Qualcomm), Frank Kastenholz (FTP Software), and Pat Calhoun (Sun Microsystems) for their generous support in hosting interim Working Group meetings.

Sections 1.10 and 1.11, which specify new extension formats to be used with aggregatable extension types, were included from a specification document (entitled "Mobile IP Extensions Rationalization (MIER)", which was written by

- Mohamed M. Khalil, Nortel Networks
- Raja Narayanan, nVisible Networks
- Haseeb Akhtar, Nortel Networks
- Emad Qaddoura, Nortel Networks

Thanks to these authors, and also for the additional work on MIER, which was contributed by Basavaraj Patil, Pat Calhoun, Neil Justusson, N. Asokan, and Jouni Malinen.

A. Patent Issues

The IETF has been notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

B. Link-Layer Considerations

The mobile node MAY use link-layer mechanisms to decide that its point of attachment has changed. Such indications include the Down/Testing/Up interface status [24], and changes in cell or administration. The mechanisms will be specific to the particular link-layer technology, and are outside the scope of this document.

The Point-to-Point-Protocol (PPP) [42] and its Internet Protocol Control Protocol (IPCP) [25], negotiates the use of IP addresses. The mobile node SHOULD first attempt to specify its home address, so that if the mobile node is attaching to its home network, the unrouted link will function correctly. When the home address is not accepted by the peer, but a transient IP address is dynamically assigned to the mobile node, and the mobile node is capable of supporting a co-located care-of address, the mobile node MAY register that address as a co-located care-of address. When the peer specifies its own IP address, that address MUST NOT be assumed to be a foreign agent care-of address or the IP address of a home agent.

PPP extensions for Mobile IP have been specified in RFC 2290 [44]. Please consult that document for additional details for how to handle care-of address assignment from PPP in a more efficient manner.

C. TCP Considerations

C.1. TCP Timers

When high-delay (e.g. SATCOM) or low-bandwidth (e.g. High-Frequency Radio) links are in use, some TCP stacks may have insufficiently adaptive (non-standard) retransmission timeouts. There may be spurious retransmission timeouts, even when the link and network are actually operating properly, but just with a high delay because of the medium in use. This can cause an inability to create or maintain TCP connections over such links, and can also cause unneeded retransmissions which consume already scarce bandwidth. Vendors are encouraged to follow the algorithms in RFC 2988 [31] when implementing TCP retransmission timers. Vendors of systems designed for low-bandwidth, high-delay links should consult RFCs 2757 and 2488 [28, 1]. Designers of applications targeted to operate on mobile nodes should be sensitive to the possibility of timer-related difficulties.

C.2. TCP Congestion Management

Mobile nodes often use media which are more likely to introduce errors, effectively causing more packets to be dropped. This introduces a conflict with the mechanisms for congestion management found in modern versions of TCP [21]. Now, when a packet is dropped, the correspondent node's TCP implementation is likely to react as if there were a source of network congestion, and initiate the slow-start mechanisms [21] designed for controlling that problem. However, those mechanisms are inappropriate for overcoming errors introduced by the links themselves, and have the effect of magnifying the discontinuity introduced by the dropped packet. This problem has been analyzed by Caceres, et al. [5]. TCP approaches to the problem of handling errors that might interfere with congestion management are discussed in documents from the [pilc] working group [3, 9]. While such approaches are beyond the scope of this document, they illustrate that providing performance transparency to mobile nodes involves understanding mechanisms outside the network layer. Problems introduced by higher media error rates also indicate the need to avoid designs which systematically drop packets; such designs might otherwise be considered favorably when making engineering tradeoffs.

D. Example Scenarios

This section shows example Registration Requests for several common scenarios.

D.1. Registering with a Foreign Agent Care-of Address

The mobile node receives an Agent Advertisement from a foreign agent and wishes to register with that agent using the advertised foreign agent care-of address. The mobile node wishes only IP-in-IP encapsulation, does not want broadcasts, and does not want simultaneous mobility bindings:

IP fields:

Source Address = mobile node's home address

Destination Address = copied from the IP source address of the Agent Advertisement

Time to Live = 1

UDP fields:

Source Port = <any>

Destination Port = 434

Registration Request fields:

Type = 1

S=0, B=0, D=0, M=0, G=0

Lifetime = the Registration Lifetime copied from the Mobility Agent Advertisement Extension of the Router Advertisement message

Home Address = the mobile node's home address

Home Agent = IP address of mobile node's home agent

Care-of Address = the Care-of Address copied from the Mobility Agent Advertisement Extension of the Router Advertisement message

Identification = Network Time Protocol timestamp or Nonce

Extensions:

An authorization-enabling extension (e.g., the Mobile-Home Authentication Extension)

D.2. Registering with a Co-Located Care-of Address

The mobile node enters a foreign network that contains no foreign agents. The mobile node obtains an address from a DHCP server [13] for use as a co-located care-of address. The mobile node supports all forms of encapsulation (IP-in-IP, minimal encapsulation, and GRE), desires a copy of broadcast datagrams on the home network, and does not want simultaneous mobility bindings:

IP fields:

Source Address = care-of address obtained from DHCP server

Destination Address = IP address of home agent

Time to Live = 64

UDP fields:

Source Port = <any>

Destination Port = 434

Registration Request fields:

Type = 1

S=0, B=1, D=1, M=1, G=1

Lifetime = 1800 (seconds)

Home Address = the mobile node's home address

Home Agent = IP address of mobile node's home agent

Care-of Address = care-of address obtained from DHCP server

Identification = Network Time Protocol timestamp or Nonce

Extensions:

The Mobile-Home Authentication Extension

D.3. Deregistration

The mobile node returns home and wishes to deregister all care-of addresses with its home agent.

IP fields:

Source Address = mobile node's home address

Destination Address = IP address of home agent

Time to Live = 1

UDP fields:

Source Port = <any>

Destination Port = 434

Registration Request fields:

Type = 1

S=0, B=0, D=0, M=0, G=0

Lifetime = 0

Home Address = the mobile node's home address

Home Agent = IP address of mobile node's home agent

Care-of Address = the mobile node's home address

Identification = Network Time Protocol timestamp or Nonce

Extensions:

An authorization-enabling extension (e.g., the Mobile-Home Authentication Extension)

E. Applicability of Prefix-Lengths Extension

Caution is indicated with the use of the Prefix-Lengths Extension over wireless links, due to the irregular coverage areas provided by wireless transmitters. As a result, it is possible that two foreign agents advertising the same prefix might indeed provide different connectivity to prospective mobile nodes. The Prefix-Lengths Extension SHOULD NOT be included in the advertisements sent by agents in such a configuration.

Foreign agents using different wireless interfaces would have to cooperate using special protocols to provide identical coverage in space, and thus be able to claim to have wireless interfaces situated on the same subnetwork. In the case of wired interfaces, a mobile node disconnecting and subsequently connecting to a new point of attachment, may well send in a new Registration Request no matter whether the new advertisement is on the same medium as the last recorded advertisement. And, finally, in areas with dense populations of foreign agents it would seem unwise to require the propagation via routing protocols of the subnet prefixes associated with each individual wireless foreign agent; such a strategy could lead to quick depletion of available space for routing tables, unwarranted increases in the time required for processing routing updates, and longer decision times for route selection if routes (which are almost always unnecessary) are stored for wireless "subnets".

F. Interoperability Considerations

This document specifies revisions to RFC 2002 that are intended to improve interoperability by resolving ambiguities contained in the earlier text. Implementations that perform authentication according to the new more precisely specified algorithm would be interoperable with earlier implementations that did what was originally expected for producing authentication data. That was a major source of non-interoperability before.

However, this specification does have new features that, if used, would cause interoperability problems with older implementations. All features specified in RFC 2002 will work with the new implementations, except for V-J compression [20]. The following list details some of the possible areas of compatibility problems that may be experienced by nodes conforming to this revised specification, when attempting to interoperate with nodes obeying RFC 2002.

- A client that expects some of the newly mandatory features (like reverse tunneling) from a foreign agent would still be interoperable as long as it pays attention to the 'T' bit.

- Mobile nodes that use the NAI extension to identify themselves would not work with old mobility agents.
- Mobile nodes that use a zero home address and expect to receive their home address in the Registration Reply would not work with old mobility agents.
- Mobile nodes that attempt to authenticate themselves without using the Mobile-Home authentication extension will be unable to successfully register with their home agent.

In all of these cases, a robust and well-configured mobile node is very likely to be able to recover if it takes reasonable actions upon receipt of a Registration Reply with an error code indicating the cause for rejection. For instance, if a mobile node sends a registration request that is rejected because it contains the wrong kind of authentication extension, then the mobile node could retry the registration with a mobile-home authentication extension, since the foreign agent and/or home agent in this case will not be configured to demand the alternative authentication data.

G. Changes since RFC 2002

This section details differences between the original Mobile IP base specification (RFC 2002 and ff.) that have been made as part of this revised protocol specification for Mobile IP.

G.1. Major Changes

- Specification for Destination IP address of Registration Reply transmitted from Foreign Agent, to avoid any possible transmission to IP address 0.0.0.0.
- Specification of two new formats for Mobile IP extensions, according to the ideas contained in MIER.
- Specification that the SPI of the MN-HA authentication extension is to be used as part of the data over which the authentication algorithm must be computed.
- Eliminated Van-Jacobson Compression feature
- Specification that foreign agents MAY send advertisements at a rate faster than once per second, but chosen so that the advertisements do not burden the capacity of the local link. For simplicity, the foreign agent now MAY send advertisements at an interval less than 1/3 the advertised ICMP Lifetime.

- Specification that foreign agents SHOULD support reverse tunneling, and home agents MUST support decapsulation of reverse tunnels.
- Changed the preconfiguration requirements in section 3.6 for the mobile node to reflect the capability, specified in RFC 2794 [6], for the mobile node to identify itself by using its NAI, and then getting a home address from the Registration Reply.
- Changed section 3.7.3.1 so that a foreign agent is not required to discard Registration Replies that have a Home Address field that does not match any pending Registration Request.
- Allowed registrations to be authenticated by use of a security association between the mobile node and a suitable authentication entity acceptable to the home agent. Defined "Authorization-enabling Extension" to be an authentication extension that makes a registration message acceptable to the recipient. This is needed according to specification in [6].
- Mandated that HMAC-MD5 be used instead of the "prefix+suffix" mode of MD5 as originally mandated in RFC 2002.
- Specified that the mobile node SHOULD take the first care-of address in a list offered by a foreign agent, and MAY try each subsequent advertised address in turn if the attempted registrations are rejected by the foreign agent
- Clarification that a mobility agent SHOULD only put its own addresses into the initial (i.e., not mobility-related) list of routers in the mobility advertisement. RFC 2002 suggests that a mobility agent might advertise other default routers.
- Specification that a mobile node MUST ignore reserved bits in Agent Advertisements, as opposed to discarding such advertisements. In this way, new bits can be defined later, without affecting the ability for mobile nodes to use the advertisements even when the newly defined bits are not understood. Furthermore, foreign agents can set the 'R' bit to make sure that new bits are handled by themselves instead of some legacy mobility agent.
- Specification that the foreign agent checks to make sure that the indicated home agent address does not belong to any of its network interfaces before relaying a Registration Request. If

the check fails, and the foreign agent is not the mobile node's home agent, then the foreign agent rejects the request with code 136 (unknown home agent address).

- Specification that, while they are away from the home network, mobile nodes **MUST NOT** broadcast ARP packets to find the MAC address of another Internet node. Thus, the (possibly empty) list of Router Addresses from the ICMP Router Advertisement portion of the message is not useful for selecting a default router, unless the mobile node has some means not involving broadcast ARP and not specified within this document for obtaining the MAC address of one of the routers in the list. Similarly, in the absence of unspecified mechanisms for obtaining MAC addresses on foreign networks, the mobile node **MUST** ignore redirects to other routers on foreign networks.
- Specification that a foreign agent **MUST NOT** use broadcast ARP for a mobile node's MAC address on a foreign network. It may obtain the MAC address by copying the information from an Agent Solicitation or a Registration Request transmitted from a mobile node.
- Specification that a foreign agent's ARP cache for the mobile node's IP address **MUST NOT** be allowed to expire before the mobile node's visitor list entry expires, unless the foreign agent has some way other than broadcast ARP to refresh its MAC address associated to the mobile node's IP address.
- At the end of section 4.6, clarified that a home agent **MUST NOT** make any changes to the way it performs proxy ARP after it rejects an invalid deregistration request.
- In section 4.2.3, specification that multihomed home agents **MUST** use the address sent to the mobile node in the home agent field of the registration reply as the source address in the outer IP header of the encapsulated datagram.
- Inserted 'T' bit into its proper place in the Registration Request message format (section 3.3).

G.2. Minor Changes

- Allowed registration replies to be processed by the mobile node, even in the absence of any Mobile-Home Authentication extension, when containing rejection code by the foreign agent.

- Specification that the foreign agent MAY configure a maximum number of pending registrations that it is willing to maintain (typically 5). Additional registrations SHOULD then be rejected by the foreign agent with code 66. The foreign agent MAY delete any pending Registration Request after the request has been pending for more than 7 seconds; in this case, the foreign agent SHOULD reject the Request with code 78 (registration timeout).
- Relaxation of the requirement that, when a mobile node has joined a multicast group at the router on the foreign network, the mobile node MUST use its home address as the source IP address for multicast packets,
- Clarification that a mobility agent MAY use different settings for each of the 'R', 'H', and 'F' bits on different network interfaces.
- Replacement of the terminology "recursive tunneling" by the terminology "nested tunneling".
- Specification that the mobile node MAY use the IP source address of an agent advertisement as its default router address.
- Clarification that keys with arbitrary binary values MUST be supported as part of mobility security associations.
- Specification that the default value may be chosen as 7 seconds, for allowable time skews between a home agent and mobile node using timestamps for replay protection. Further specification that this value SHOULD be greater than 3 seconds.
- Specification that Registration Requests with the 'D' bit set to 0, and specifying a care-of address not offered by the foreign agent, MUST be rejected with code 77 (invalid care-of address).
- Clarification that the foreign agent SHOULD consider its own maximum value when handling the Lifetime field of the Registration Reply.
- Clarification that the home agent MUST ignore the 'B' bit (as opposed to rejecting the Registration Request) if it does not support broadcasts.

- Advice about the impossibility of using dynamic home agent discovery in the case when routers change the IP destination address of a datagram from a subnet-directed broadcast address to 255.255.255.255 before injecting it into the destination subnet.
- Clarified that when an Agent Advertisement is unicast to a mobile node, the specific IP home address of a mobile node MAY be used as the destination IP address.
- Included a reference to RFC 2290 within appendix B, which deals with PPP operation.
- Created IANA Considerations section
- In section 3.8.3, clarified that a home agent SHOULD arrange the selection of a home address for a mobile node when the Registration Reply contains a zero Home Address.

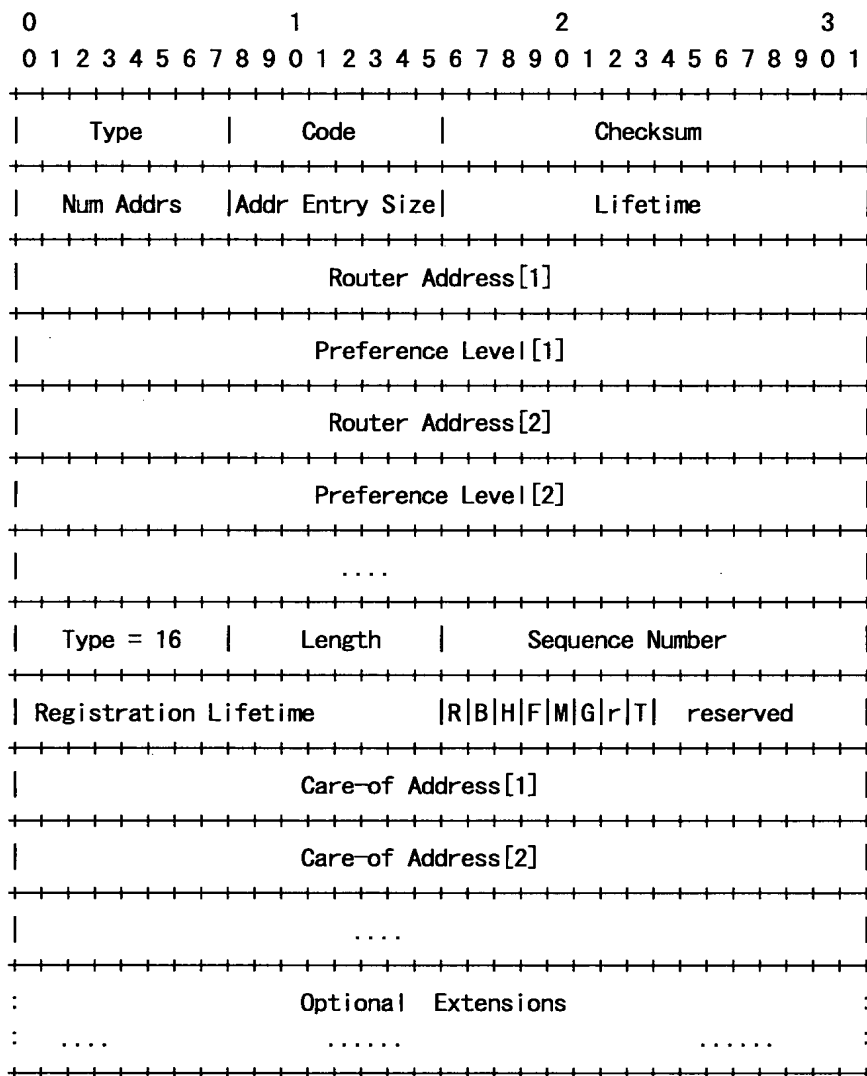
G.3. Changes since revision 04 of RFC2002bis

This section lists the changes between this version (...-06.txt) and the previous version (...-04.txt) of the document. This section can be deleted by the RFC editor.

- Noted that HMAC-MD5 should be considered for use in place of the "prefix+suffix" mode of MD5 as originally mandated in RFC 2002.
- Included a reference to RFC 2290 within appendix B, which deals with PPP operation.
- Revamped IANA Considerations section
- Revamped Changes section
- Replaced Patents section with wording mandated from RFC 2026.
- Updated citations.

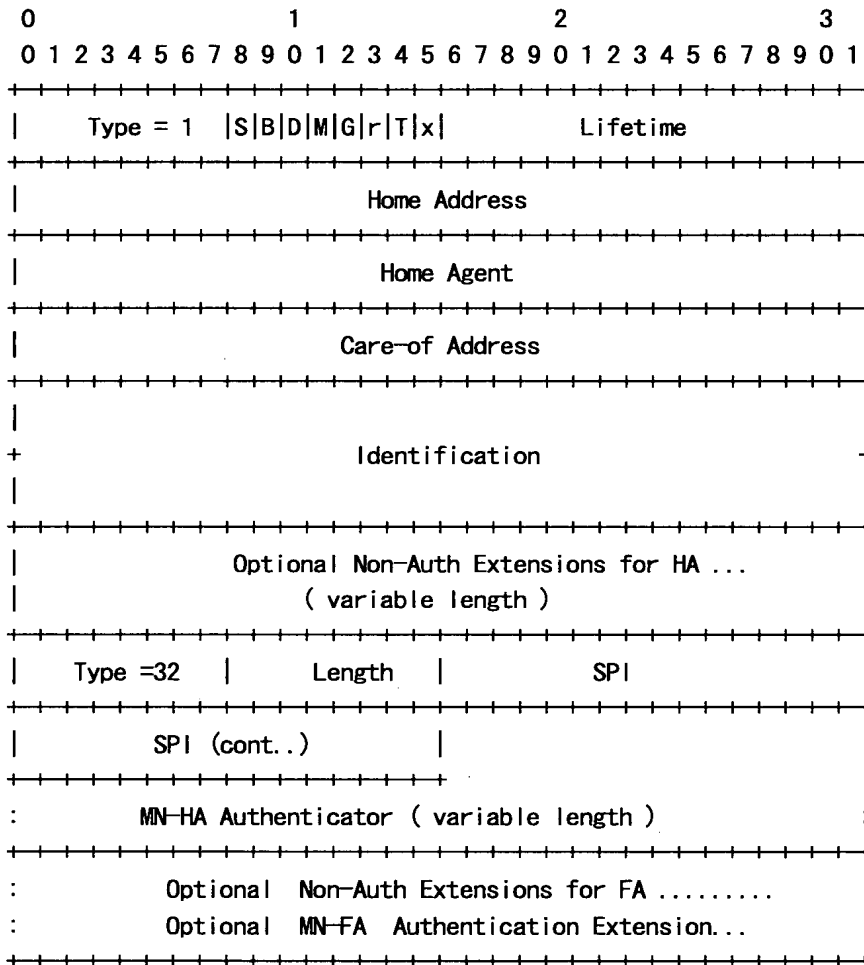
H. Example Messages

H.1. Example ICMP Agent Advertisement Message Format



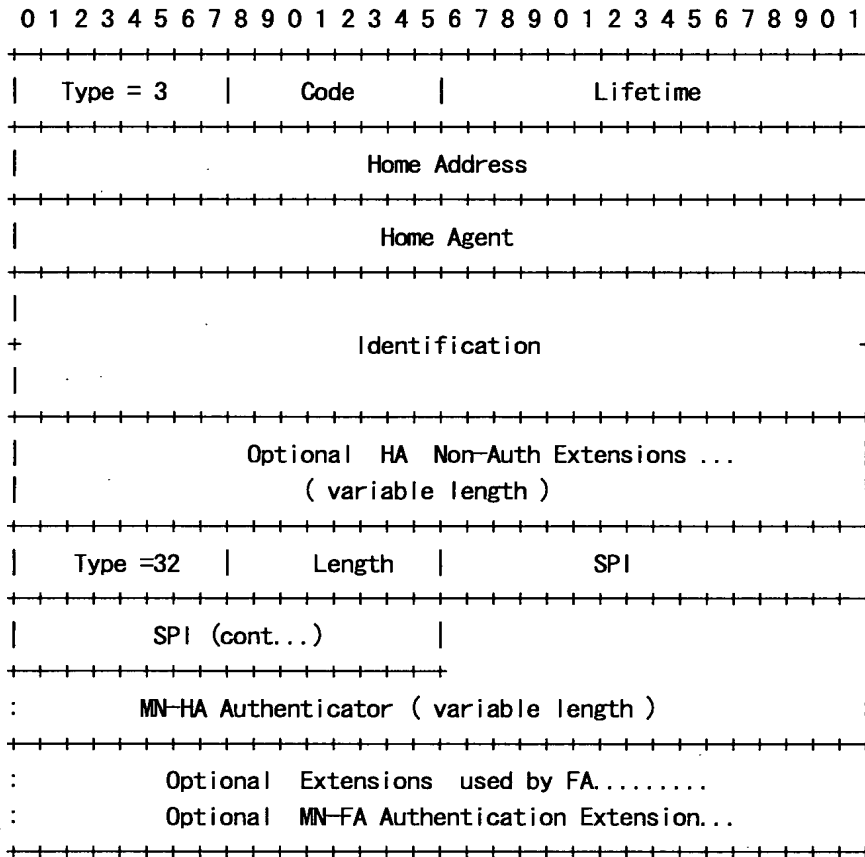
H.2. Example Registration Request Message Format

The UDP header is followed by the Mobile IP fields shown below:



H.3. Example Registration Reply Message Format

The UDP header is followed by the Mobile IP fields shown below:



References

- [1] Allman, M., Glover, D. and L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", BCP 28, RFC 2488, January 1999.
- [2] S. M. Bellovin. Security Problems in the TCP/IP Protocol Suite. ACM Computer Communications Review, 19(2), March 1989.
- [3] Border, J., Kojo, M., Griner, J., Montenegro, G. and Z. Shelby, "Performance Enhancing Proxies", RFC 3135, June 2001.
- [4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [5] Ramon Caceres and Liviu Iftode. Improving the Performance of Reliable Transport Protocols in Mobile Computing Environments. IEEE Journal on Selected Areas in Communications, 13(5):850—857, June 1995.
- [6] Calhoun P. and C. Perkins, "Mobile IP Network Access Identifier Extension for IPv4", RFC 2794, January 2000.
- [7] Calhoun, P. and C. Perkins, "Mobile IP Foreign Agent Challenge/Response Extension", RFC 3012, December 2000.
- [8] Cong, D., Hamlen, M. and C. Perkins, "The Definitions of Managed Objects for IP Mobility Support using SMIv2", RFC 2006, October 1996.
- [9] Dawkins, S., Montenegro, G., Kojo, M., Magret, V. and N. Vaidya, "End-to-end Performance Implications of Links with Errors", BCP 50, RFC 3155, August 2001.
- [10] Deering, S., "ICMP Router Discovery Messages", RFC 1256, September 1991.
- [11] Deering, S., "Host Extensions for IP Multicasting", STD 5, RFC 1112, August 1989.
- [12] Domety, G. and K. Leung, "Mobile IP Vendor/Organization-Specific Extensions", RFC 3115, April 2001.
- [13] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.
- [14] Eastlake, D., Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.
- [15] Ferguson P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.
- [16] Hanks, S., Li, T., Farinacci, D. and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 1701, October 1994.
- [17] J. Ioannidis. Protocols for Mobile Internetworking. PhD Dissertation - Columbia University in the City of New York, July 1993.

- [18] John Ioannidis, Dan Duchamp, and Gerald Q. Maguire Jr. IP-Based Protocols for Mobile Internetworking. In Proceedings of the SIGCOMM '91 Conference: Communications Architectures & Protocols, pages 235—245, September 1991.
- [19] John Ioannidis and Gerald Q. Maguire Jr. The Design and Implementation of a Mobile Internetworking Architecture. In Proceedings of the Winter USENIX Technical Conference, pages 489—500, January 1993.
- [20] Jacobson, V., "Compressing TCP/IP headers for low-speed serial links", RFC 1144, February 1990.
- [21] Jacobson, V., "Congestion Avoidance and Control. In Proceedings, SIGCOMM '88 Workshop, pages 314—329. ACM Press, August 1988. Stanford, CA.
- [22] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.
- [23] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.
- [24] McCloghrie, K. and F. Kastenholtz, "The Interfaces Group MIB", RFC 2863, June 2000.
- [25] McGregor, G., "The PPP Internet Protocol Control Protocol (IPCP)", RFC 1332, May 1992.
- [26] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", RFC 1305, March 1992.
- [27] Montenegro, G., "Reverse Tunneling for Mobile IP (revised)", RFC 3024, January 2001.
- [28] Montenegro, G., Dawkins, S., Kojo, M., Magret, V. and N. Vaidya, "Long Thin Networks", RFC 2757, January 2000.
- [29] Montenegro, G. and V. Gupta, "Sun's SKIP Firewall Traversal for Mobile IP", RFC 2356, June 1998.
- [30] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, October 1998.
- [31] Paxson, V. and M. Allman, "Computing TCP's Retransmission Timer", RFC 2988, November 2000.

- [32] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.
- [33] Perkins, C., "IP Mobility Support", RFC 2002, October 1996.
- [34] Perkins, C., "Minimal Encapsulation within IP", RFC 2004, October 1996.
- [35] Perkins, C. and P. Calhoun, "AAA Registration Keys for Mobile IP", Work in Progress, July 2001.
- [36] Plummer, D., "Ethernet Address Resolution Protocol: Or converting network protocol addresses to 48.bit Ethernet address for transmission on Ethernet hardware", STD 37, RFC 826, November 1982.
- [37] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.
- [38] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [39] Postel, J., "Multi-LAN Address Resolution", RFC 925, October 1984.
- [40] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994.
- [41] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992.
- [42] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.
- [43] Solomon, J., "Applicability Statement for IP Mobility Support" RFC 2005, October 1996.
- [44] Solomon J. and S. Glass, "Mobile-IPv4 Configuration Option for PPP IPCP", RFC 2290, February 1998.
- [45] Stevens, W., "TCP/IP Illustrated, Volume 1: The Protocols" Addison-Wesley, Reading, Massachusetts, 1994.

Authors' Addresses

The working group can be contacted via the current chairs:

Basavaraj Patil
Nokia
6000 Connection Dr.
Irving, TX. 75039
USA

Phone: +1 972-894-6709
EMail: Basavaraj.Patil@nokia.com

Phil Roberts
Megisto Corp. Suite 120
20251 Century Blvd
Germantown MD 20874
USA

Phone: +1 847-202-9314
EMail: PROberts@MEGISTO.com

Questions about this memo can also be directed to the editor:

Charles E. Perkins
Communications Systems Lab
Nokia Research Center
313 Fairchild Drive
Mountain View, California 94043
USA

Phone: +1-650 625-2986
EMail: charliep@iprg.nokia.com
Fax: +1 650 625-2502

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

INTERNET PROTOCOL
DARPA INTERNET PROGRAM
PROTOCOL SPECIFICATION

September 1981

prepared for

Defense Advanced Research Projects Agency
Information Processing Techniques Office
1400 Wilson Boulevard
Arlington, Virginia 22209

by

Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, California 90291

TABLE OF CONTENTS

PREFACE	iii
1. INTRODUCTION	1
1.1 Motivation	1
1.2 Scope	1
1.3 Interfaces	1
1.4 Operation	2
2. OVERVIEW	5
2.1 Relation to Other Protocols	9
2.2 Model of Operation	5
2.3 Function Description	7
2.4 Gateways	9
3. SPECIFICATION	11
3.1 Internet Header Format	11
3.2 Discussion	23
3.3 Interfaces	31
APPENDIX A: Examples & Scenarios	34
APPENDIX B: Data Transmission Order	39
GLOSSARY	41
REFERENCES	45

PREFACE

This document specifies the DoD Standard Internet Protocol. This document is based on six earlier editions of the ARPA Internet Protocol Specification, and the present text draws heavily from them. There have been many contributors to this work both in terms of concepts and in terms of text. This edition revises aspects of addressing, error handling, option codes, and the security, precedence, compartments, and handling restriction features of the internet protocol.

Jon Postel

Editor

RFC: 791

Replaces: RFC 760

IENs 128, 123, 111,

80, 54, 44, 41, 28, 26

INTERNET PROTOCOL

DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION

1. INTRODUCTION

1.1. Motivation

The Internet Protocol is designed for use in interconnected systems of packet-switched computer communication networks. Such a system has been called a "catenet" [1]. The internet protocol provides for transmitting blocks of data called datagrams from sources to destinations, where sources and destinations are hosts identified by fixed length addresses. The internet protocol also provides for fragmentation and reassembly of long datagrams, if necessary, for transmission through "small packet" networks.

1.2. Scope

The internet protocol is specifically limited in scope to provide the functions necessary to deliver a package of bits (an internet datagram) from a source to a destination over an interconnected system of networks. There are no mechanisms to augment end-to-end data reliability, flow control, sequencing, or other services commonly found in host-to-host protocols. The internet protocol can capitalize on the services of its supporting networks to provide various types and qualities of service.

1.3. Interfaces

This protocol is called on by host-to-host protocols in an internet environment. This protocol calls on local network protocols to carry the internet datagram to the next gateway or destination host.

For example, a TCP module would call on the internet module to take a TCP segment (including the TCP header and user data) as the data portion of an internet datagram. The TCP module would provide the addresses and other parameters in the internet header to the internet module as arguments of the call. The internet module would then create an internet datagram and call on the local network interface to transmit the internet datagram.

In the ARPANET case, for example, the internet module would call on a

Internet Protocol Introduction

local net module which would add the 1822 leader [2] to the internet datagram creating an ARPANET message to transmit to the IMP. The ARPANET address would be derived from the internet address by the local network interface and would be the address of some host in the ARPANET, that host might be a gateway to other networks.

1.4. Operation

The internet protocol implements two basic functions: addressing and fragmentation.

The internet modules use the addresses carried in the internet header to transmit internet datagrams toward their destinations. The selection of a path for transmission is called routing.

The internet modules use fields in the internet header to fragment and reassemble internet datagrams when necessary for transmission through "small packet" networks.

The model of operation is that an internet module resides in each host engaged in internet communication and in each gateway that interconnects networks. These modules share common rules for interpreting address fields and for fragmenting and assembling internet datagrams. In addition, these modules (especially in gateways) have procedures for making routing decisions and other functions.

The internet protocol treats each internet datagram as an independent entity unrelated to any other internet datagram. There are no connections or logical circuits (virtual or otherwise).

The internet protocol uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

The Type of Service is used to indicate the quality of the service desired. The type of service is an abstract or generalized set of parameters which characterize the service choices provided in the networks that make up the internet. This type of service indication is to be used by gateways to select the actual transmission parameters for a particular network, the network to be used for the next hop, or the next gateway when routing an internet datagram.

The Time to Live is an indication of an upper bound on the lifetime of an internet datagram. It is set by the sender of the datagram and reduced at the points along the route where it is processed. If the time to live reaches zero before the internet datagram reaches its destination, the internet datagram is destroyed. The time to live can be thought of as a self destruct time limit.

The Options provide for control functions needed or useful in some situations but unnecessary for the most common communications. The options include provisions for timestamps, security, and special routing.

The Header Checksum provides a verification that the information used in processing internet datagram has been transmitted correctly. The data may contain errors. If the header checksum fails, the internet datagram is discarded at once by the entity which detects the error.

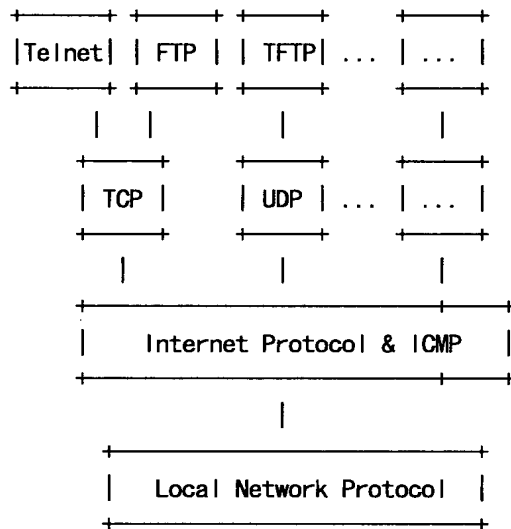
The internet protocol does not provide a reliable communication facility. There are no acknowledgments either end-to-end or hop-by-hop. There is no error control for data, only a header checksum. There are no retransmissions. There is no flow control.

Errors detected may be reported via the Internet Control Message Protocol (ICMP) [3] which is implemented in the internet protocol module.

2. OVERVIEW

2.1. Relation to Other Protocols

The following diagram illustrates the place of the internet protocol in the protocol hierarchy:



Protocol Relationships

Figure 1.

Internet protocol interfaces on one side to the higher level host-to-host protocols and on the other side to the local network protocol. In this context a "local network" may be a small network in a building or a large network such as the ARPANET.

2.2. Model of Operation

The model of operation for transmitting a datagram from one application program to another is illustrated by the following scenario:

We suppose that this transmission will involve one intermediate gateway.

The sending application program prepares its data and calls on its local internet module to send that data as a datagram and passes the destination address and other parameters as arguments of the call.

The internet module prepares a datagram header and attaches the data to it. The internet module determines a local network address for this internet address, in this case it is the address of a gateway.

Internet Protocol Overview

It sends this datagram and the local network address to the local network interface.

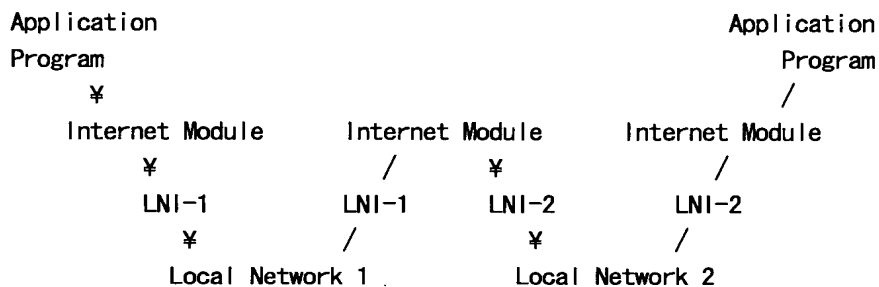
The local network interface creates a local network header, and attaches the datagram to it, then sends the result via the local network.

The datagram arrives at a gateway host wrapped in the local network header, the local network interface strips off this header, and turns the datagram over to the internet module. The internet module determines from the internet address that the datagram is to be forwarded to another host in a second network. The internet module determines a local net address for the destination host. It calls on the local network interface for that network to send the datagram.

This local network interface creates a local network header and attaches the datagram sending the result to the destination host.

At this destination host the datagram is stripped of the local net header by the local network interface and handed to the internet module.

The internet module determines that the datagram is for an application program in this host. It passes the data to the application program in response to a system call, passing the source address and other parameters as results of the call.



Transmission Path

Figure 2

2.3. Function Description

The function or purpose of Internet Protocol is to move datagrams through an interconnected set of networks. This is done by passing the datagrams from one internet module to another until the destination is reached. The internet modules reside in hosts and gateways in the internet system. The datagrams are routed from one internet module to another through individual networks based on the interpretation of an internet address. Thus, one important mechanism of the internet protocol is the internet address.

In the routing of messages from one internet module to another, datagrams may need to traverse a network whose maximum packet size is smaller than the size of the datagram. To overcome this difficulty, a fragmentation mechanism is provided in the internet protocol.

Addressing

A distinction is made between names, addresses, and routes [4]. A name indicates what we seek. An address indicates where it is. A route indicates how to get there. The internet protocol deals primarily with addresses. It is the task of higher level (i.e., host-to-host or application) protocols to make the mapping from names to addresses. The internet module maps internet addresses to local net addresses. It is the task of lower level (i.e., local net or gateways) procedures to make the mapping from local net addresses to routes.

Addresses are fixed length of four octets (32 bits). An address begins with a network number, followed by local address (called the "rest" field). There are three formats or classes of internet addresses: in class a, the high order bit is zero, the next 7 bits are the network, and the last 24 bits are the local address; in class b, the high order two bits are one-zero, the next 14 bits are the network and the last 16 bits are the local address; in class c, the high order three bits are one-one-zero, the next 21 bits are the network and the last 8 bits are the local address.

Care must be taken in mapping internet addresses to local net addresses: a single physical host must be able to act as if it were several distinct hosts to the extent of using several distinct internet addresses. Some hosts will also have several physical interfaces (multi-homing).

That is, provision must be made for a host to have several physical interfaces to the network with each having several logical internet addresses.

Internet Protocol Overview

Examples of address mappings may be found in "Address Mappings" [5].

Fragmentation

Fragmentation of an internet datagram is necessary when it originates in a local net that allows a large packet size and must traverse a local net that limits packets to a smaller size to reach its destination.

An internet datagram can be marked "don't fragment." Any internet datagram so marked is not to be internet fragmented under any circumstances. If internet datagram marked don't fragment cannot be delivered to its destination without fragmenting it, it is to be discarded instead.

Fragmentation, transmission and reassembly across a local network which is invisible to the internet protocol module is called intranet fragmentation and may be used [6].

The internet fragmentation and reassembly procedure needs to be able to break a datagram into an almost arbitrary number of pieces that can be later reassembled. The receiver of the fragments uses the identification field to ensure that fragments of different datagrams are not mixed. The fragment offset field tells the receiver the position of a fragment in the original datagram. The fragment offset and length determine the portion of the original datagram covered by this fragment. The more-fragments flag indicates (by being reset) the last fragment. These fields provide sufficient information to reassemble datagrams.

The identification field is used to distinguish the fragments of one datagram from those of another. The originating protocol module of an internet datagram sets the identification field to a value that must be unique for that source-destination pair and protocol for the time the datagram will be active in the internet system. The originating protocol module of a complete datagram sets the more-fragments flag to zero and the fragment offset to zero.

To fragment a long internet datagram, an internet protocol module (for example, in a gateway), creates two new internet datagrams and copies the contents of the internet header fields from the long datagram into both new internet headers. The data of the long datagram is divided into two portions on a 8 octet (64 bit) boundary (the second portion might not be an integral multiple of 8 octets, but the first must be). Call the number of 8 octet blocks in the first portion NFB (for Number of Fragment Blocks). The first portion of the data is placed in the first new internet datagram, and the total length field is set to the length of the first

datagram. The more-fragments flag is set to one. The second portion of the data is placed in the second new internet datagram, and the total length field is set to the length of the second datagram. The more-fragments flag carries the same value as the long datagram. The fragment offset field of the second new internet datagram is set to the value of that field in the long datagram plus NFB.

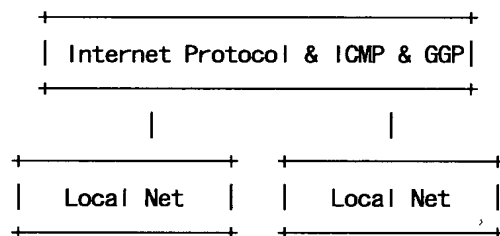
This procedure can be generalized for an n-way split, rather than the two-way split described.

To assemble the fragments of an internet datagram, an internet protocol module (for example at a destination host) combines internet datagrams that all have the same value for the four fields: identification, source, destination, and protocol. The combination is done by placing the data portion of each fragment in the relative position indicated by the fragment offset in that fragment's internet header. The first fragment will have the fragment offset zero, and the last fragment will have the more-fragments flag reset to zero.

2.4. Gateways

Gateways implement internet protocol to forward datagrams between networks. Gateways also implement the Gateway to Gateway Protocol (GGP) [7] to coordinate routing and other internet control information.

In a gateway the higher level protocols need not be implemented and the GGP functions are added to the IP module.



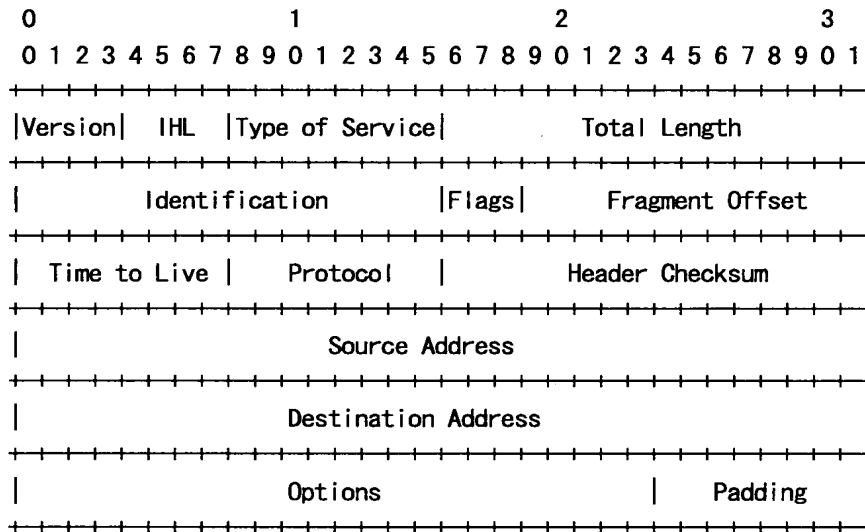
Gateway Protocols

Figure 3.

3. SPECIFICATION

3.1. Internet Header Format

A summary of the contents of the internet header follows:



Example Internet Datagram Header

Figure 4.

Note that each tick mark represents one bit position.

Version: 4 bits

The Version field indicates the format of the internet header. This document describes version 4.

IHL: 4 bits

Internet Header Length is the length of the internet header in 32 bit words, and thus points to the beginning of the data. Note that the minimum value for a correct header is 5.

Internet Protocol Specification

Type of Service: 8 bits

The Type of Service provides an indication of the abstract parameters of the quality of service desired. These parameters are to be used to guide the selection of the actual service parameters when transmitting a datagram through a particular network. Several networks offer service precedence, which somehow treats high precedence traffic as more important than other traffic (generally by accepting only traffic above a certain precedence at time of high load). The major choice is a three way tradeoff between low-delay, high-reliability, and high-throughput.

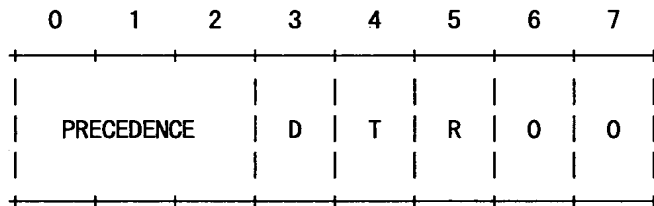
Bits 0-2: Precedence.

Bit 3: 0 = Normal Delay, 1 = Low Delay.

Bits 4: 0 = Normal Throughput, 1 = High Throughput.

Bits 5: 0 = Normal Reliability, 1 = High Reliability.

Bit 6-7: Reserved for Future Use.



Precedence

- 111 - Network Control
- 110 - Internetwork Control
- 101 - CRITIC/ECP
- 100 - Flash Override
- 011 - Flash
- 010 - Immediate
- 001 - Priority
- 000 - Routine

The use of the Delay, Throughput, and Reliability indications may increase the cost (in some sense) of the service. In many networks better performance for one of these parameters is coupled with worse performance on another. Except for very unusual cases at most two of these three indications should be set.

The type of service is used to specify the treatment of the datagram during its transmission through the internet system. Example mappings of the internet type of service to the actual service provided on networks such as AUTODIN II, ARPANET, SATNET, and PRNET is given in "Service Mappings" [8].

The Network Control precedence designation is intended to be used within a network only. The actual use and control of that designation is up to each network. The Internetwork Control designation is intended for use by gateway control originators only. If the actual use of these precedence designations is of concern to a particular network, it is the responsibility of that network to control the access to, and use of, those precedence designations.

Total Length: 16 bits

Total Length is the length of the datagram, measured in octets, including internet header and data. This field allows the length of a datagram to be up to 65,535 octets. Such long datagrams are impractical for most hosts and networks. All hosts must be prepared to accept datagrams of up to 576 octets (whether they arrive whole or in fragments). It is recommended that hosts only send datagrams larger than 576 octets if they have assurance that the destination is prepared to accept the larger datagrams.

The number 576 is selected to allow a reasonable sized data block to be transmitted in addition to the required header information. For example, this size allows a data block of 512 octets plus 64 header octets to fit in a datagram. The maximal internet header is 60 octets, and a typical internet header is 20 octets, allowing a margin for headers of higher level protocols.

Identification: 16 bits

An identifying value assigned by the sender to aid in assembling the fragments of a datagram.

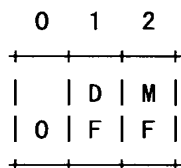
Flags: 3 bits

Various Control Flags.

Bit 0: reserved, must be zero

Bit 1: (DF) 0 = May Fragment, 1 = Don't Fragment.

Bit 2: (MF) 0 = Last Fragment, 1 = More Fragments.



Fragment Offset: 13 bits

This field indicates where in the datagram this fragment belongs.

Internet Protocol
Specification

The fragment offset is measured in units of 8 octets (64 bits). The first fragment has offset zero.

Time to Live: 8 bits

This field indicates the maximum time the datagram is allowed to remain in the internet system. If this field contains the value zero, then the datagram must be destroyed. This field is modified in internet header processing. The time is measured in units of seconds, but since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.

Protocol: 8 bits

This field indicates the next level protocol used in the data portion of the internet datagram. The values for various protocols are specified in "Assigned Numbers" [9].

Header Checksum: 16 bits

A checksum on the header only. Since some header fields change (e.g., time to live), this is recomputed and verified at each point that the internet header is processed.

The checksum algorithm is:

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of the checksum field is zero.

This is a simple to compute checksum and experimental evidence indicates it is adequate, but it is provisional and may be replaced by a CRC procedure, depending on further experience.

Source Address: 32 bits

The source address. See section 3.2.

Destination Address: 32 bits

The destination address. See section 3.2.

Options: variable

The options may appear or not in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation.

In some environments the security option may be required in all datagrams.

The option field is variable in length. There may be zero or more options. There are two cases for the format of an option:

Case 1: A single octet of option-type.

Case 2: An option-type octet, an option-length octet, and the actual option-data octets.

The option-length octet counts the option-type octet and the option-length octet as well as the option-data octets.

The option-type octet is viewed as having 3 fields:

- 1 bit copied flag,
- 2 bits option class,
- 5 bits option number.

The copied flag indicates that this option is copied into all fragments on fragmentation.

- 0 = not copied
- 1 = copied

The option classes are:

- 0 = control
- 1 = reserved for future use
- 2 = debugging and measurement
- 3 = reserved for future use

Internet Protocol Specification

The following internet options are defined:

CLASS	NUMBER	LENGTH	DESCRIPTION
0	0	-	End of Option list. This option occupies only 1 octet; it has no length octet.
0	1	-	No Operation. This option occupies only 1 octet; it has no length octet.
0	2	11	Security. Used to carry Security, Compartmentation, User Group (TCC), and Handling Restriction Codes compatible with DOD requirements.
0	3	var.	Loose Source Routing. Used to route the internet datagram based on information supplied by the source.
0	9	var.	Strict Source Routing. Used to route the internet datagram based on information supplied by the source.
0	7	var.	Record Route. Used to trace the route an internet datagram takes.
0	8	4	Stream ID. Used to carry the stream identifier.
2	4	var.	Internet Timestamp.

Specific Option Definitions

End of Option List

```

+-----+
|00000000|
+-----+
Type=0

```

This option indicates the end of the option list. This might not coincide with the end of the internet header according to the internet header length. This is used at the end of all options, not the end of each option, and need only be used if the end of the options would not otherwise coincide with the end of the internet header.

May be copied, introduced, or deleted on fragmentation, or for any other reason.

No Operation

```

+-----+
|00000001|
+-----+
Type=1

```

This option may be used between options, for example, to align the beginning of a subsequent option on a 32 bit boundary.

May be copied, introduced, or deleted on fragmentation, or for any other reason.

Security

This option provides a way for hosts to send security, compartmentation, handling restrictions, and TCC (closed user group) parameters. The format for this option is as follows:

```

+-----+-----+//+-----+//+-----+//+-----+//+
|10000010|00001011|SSS SSS|CCC CCC|HHH HHH| TCC |
+-----+-----+//+-----+//+-----+//+-----+//+
Type=130 Length=11

```

Security (S field): 16 bits

Specifies one of 16 levels of security (eight of which are reserved for future use).

```

00000000 00000000 - Unclassified
11110001 00110101 - Confidential
01111000 10011010 - EFTO
10111100 01001101 - MMMM
01011110 00100110 - PROG
10101111 00010011 - Restricted
11010111 10001000 - Secret
01101011 11000101 - Top Secret
00110101 11100010 - (Reserved for future use)
10011010 11110001 - (Reserved for future use)
01001101 01111000 - (Reserved for future use)
00100100 10111101 - (Reserved for future use)
00010011 01011110 - (Reserved for future use)
10001001 10101111 - (Reserved for future use)
11000100 11010110 - (Reserved for future use)
11100010 01101011 - (Reserved for future use)

```

Internet Protocol Specification

Compartments (C field): 16 bits

An all zero value is used when the information transmitted is not compartmented. Other values for the compartments field may be obtained from the Defense Intelligence Agency.

Handling Restrictions (H field): 16 bits

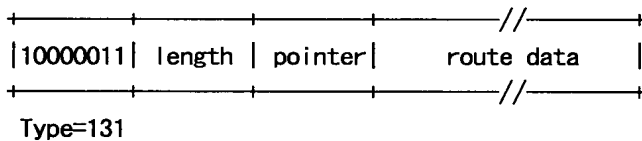
The values for the control and release markings are alphanumeric digraphs and are defined in the Defense Intelligence Agency Manual DIAM 65-19, "Standard Security Markings".

Transmission Control Code (TCC field): 24 bits

Provides a means to segregate traffic and define controlled communities of interest among subscribers. The TCC values are trigraphs, and are available from HQ DCA Code 530.

Must be copied on fragmentation. This option appears at most once in a datagram.

Loose Source and Record Route



The loose source and record route (LSRR) option provides a means for the source of an internet datagram to supply routing information to be used by the gateways in forwarding the datagram to the destination, and to record the route information.

The option begins with the option type code. The second octet is the option length which includes the option type code and the length octet, the pointer octet, and length-3 octets of route data. The third octet is the pointer into the route data indicating the octet which begins the next source address to be processed. The pointer is relative to this option, and the smallest legal value for the pointer is 4.

A route data is composed of a series of internet addresses. Each internet address is 32 bits or 4 octets. If the pointer is greater than the length, the source route is empty (and the recorded route full) and the routing is to be based on the destination address field.

If the address in destination address field has been reached and the pointer is not greater than the length, the next address in the source route replaces the address in the destination address field, and the recorded route address replaces the source address just used, and pointer is increased by four.

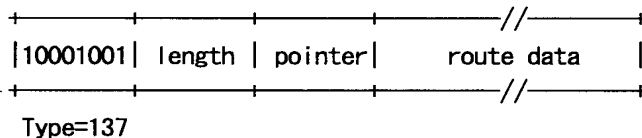
The recorded route address is the internet module's own internet address as known in the environment into which this datagram is being forwarded.

This procedure of replacing the source route with the recorded route (though it is in the reverse of the order it must be in to be used as a source route) means the option (and the IP header as a whole) remains a constant length as the datagram progresses through the internet.

This option is a loose source route because the gateway or host IP is allowed to use any route of any number of other intermediate gateways to reach the next address in the route.

Must be copied on fragmentation. Appears at most once in a datagram.

Strict Source and Record Route



The strict source and record route (SSRR) option provides a means for the source of an internet datagram to supply routing information to be used by the gateways in forwarding the datagram to the destination, and to record the route information.

The option begins with the option type code. The second octet is the option length which includes the option type code and the length octet, the pointer octet, and length-3 octets of route data. The third octet is the pointer into the route data indicating the octet which begins the next source address to be processed. The pointer is relative to this option, and the smallest legal value for the pointer is 4.

A route data is composed of a series of internet addresses. Each internet address is 32 bits or 4 octets. If the pointer is greater than the length, the source route is empty (and the

recorded route full) and the routing is to be based on the destination address field.

If the address in destination address field has been reached and the pointer is not greater than the length, the next address in the source route replaces the address in the destination address field, and the recorded route address replaces the source address just used, and pointer is increased by four.

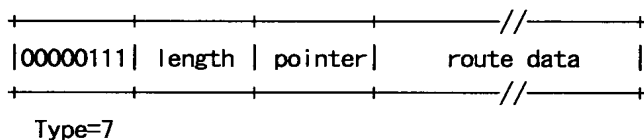
The recorded route address is the internet module's own internet address as known in the environment into which this datagram is being forwarded.

This procedure of replacing the source route with the recorded route (though it is in the reverse of the order it must be in to be used as a source route) means the option (and the IP header as a whole) remains a constant length as the datagram progresses through the internet.

This option is a strict source route because the gateway or host IP must send the datagram directly to the next address in the source route through only the directly connected network indicated in the next address to reach the next gateway or host specified in the route.

Must be copied on fragmentation. Appears at most once in a datagram.

Record Route



The record route option provides a means to record the route of an internet datagram.

The option begins with the option type code. The second octet is the option length which includes the option type code and the length octet, the pointer octet, and length-3 octets of route data. The third octet is the pointer into the route data indicating the octet which begins the next area to store a route address. The pointer is relative to this option, and the smallest legal value for the pointer is 4.

A recorded route is composed of a series of internet addresses. Each internet address is 32 bits or 4 octets. If the pointer is

greater than the length, the recorded route data area is full. The originating host must compose this option with a large enough route data area to hold all the address expected. The size of the option does not change due to adding addresses. The initial contents of the route data area must be zero.

When an internet module routes a datagram it checks to see if the record route option is present. If it is, it inserts its own internet address as known in the environment into which this datagram is being forwarded into the recorded route beginning at the octet indicated by the pointer, and increments the pointer by four.

If the route data area is already full (the pointer exceeds the length) the datagram is forwarded without inserting the address into the recorded route. If there is some room but not enough room for a full address to be inserted, the original datagram is considered to be in error and is discarded. In either case an ICMP parameter problem message may be sent to the source host [3].

Not copied on fragmentation, goes in first fragment only.
Appears at most once in a datagram.

Stream Identifier

```

+-----+-----+-----+-----+
|10001000|00000010|   Stream ID   |
+-----+-----+-----+-----+
Type=136 Length=4

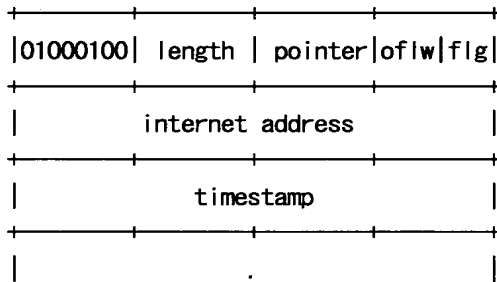
```

This option provides a way for the 16-bit SATNET stream identifier to be carried through networks that do not support the stream concept.

Must be copied on fragmentation. Appears at most once in a datagram.

Internet Protocol Specification

Internet Timestamp



Type = 68

The Option Length is the number of octets in the option counting the type, length, pointer, and overflow/flag octets (maximum length 40).

The Pointer is the number of octets from the beginning of this option to the end of timestamps plus one (i.e., it points to the octet beginning the space for next timestamp). The smallest legal value is 5. The timestamp area is full when the pointer is greater than the length.

The Overflow (oflw) [4 bits] is the number of IP modules that cannot register timestamps due to lack of space.

The Flag (flg) [4 bits] values are

- 0 — time stamps only, stored in consecutive 32-bit words,
- 1 — each timestamp is preceded with internet address of the registering entity,
- 3 — the internet address fields are prespecified. An IP module only registers its timestamp if it matches its own address with the next specified internet address.

The Timestamp is a right-justified, 32-bit timestamp in milliseconds since midnight UT. If the time is not available in milliseconds or cannot be provided with respect to midnight UT then any time may be inserted as a timestamp provided the high order bit of the timestamp field is set to one to indicate the use of a non-standard value.

The originating host must compose this option with a large enough timestamp data area to hold all the timestamp information expected. The size of the option does not change due to adding

timestamps. The initial contents of the timestamp data area must be zero or internet address/zero pairs.

If the timestamp data area is already full (the pointer exceeds the length) the datagram is forwarded without inserting the timestamp, but the overflow count is incremented by one.

If there is some room but not enough room for a full timestamp to be inserted, or the overflow count itself overflows, the original datagram is considered to be in error and is discarded. In either case an ICMP parameter problem message may be sent to the source host [3].

The timestamp option is not copied upon fragmentation. It is carried in the first fragment. Appears at most once in a datagram.

Padding: variable

The internet header padding is used to ensure that the internet header ends on a 32 bit boundary. The padding is zero.

3.2. Discussion

The implementation of a protocol must be robust. Each implementation must expect to interoperate with others created by different individuals. While the goal of this specification is to be explicit about the protocol there is the possibility of differing interpretations. In general, an implementation must be conservative in its sending behavior, and liberal in its receiving behavior. That is, it must be careful to send well-formed datagrams, but must accept any datagram that it can interpret (e.g., not object to technical errors where the meaning is still clear).

The basic internet service is datagram oriented and provides for the fragmentation of datagrams at gateways, with reassembly taking place at the destination internet protocol module in the destination host. Of course, fragmentation and reassembly of datagrams within a network or by private agreement between the gateways of a network is also allowed since this is transparent to the internet protocols and the higher-level protocols. This transparent type of fragmentation and reassembly is termed "network-dependent" (or intranet) fragmentation and is not discussed further here.

Internet addresses distinguish sources and destinations to the host level and provide a protocol field as well. It is assumed that each protocol will provide for whatever multiplexing is necessary within a host.

Internet Protocol Specification

Addressing

To provide for flexibility in assigning address to networks and allow for the large number of small to intermediate sized networks the interpretation of the address field is coded to specify a small number of networks with a large number of host, a moderate number of networks with a moderate number of hosts, and a large number of networks with a small number of hosts. In addition there is an escape code for extended addressing mode.

Address Formats:

High Order Bits	Format	Class
0	7 bits of net, 24 bits of host	a
10	14 bits of net, 16 bits of host	b
110	21 bits of net, 8 bits of host	c
111	escape to extended addressing mode	

A value of zero in the network field means this network. This is only used in certain ICMP messages. The extended addressing mode is undefined. Both of these features are reserved for future use.

The actual values assigned for network addresses is given in "Assigned Numbers" [9].

The local address, assigned by the local network, must allow for a single physical host to act as several distinct internet hosts. That is, there must be a mapping between internet host addresses and network/host interfaces that allows several internet addresses to correspond to one interface. It must also be allowed for a host to have several physical interfaces and to treat the datagrams from several of them as if they were all addressed to a single host.

Address mappings between internet addresses and addresses for ARPANET, SATNET, PRNET, and other networks are described in "Address Mappings" [5].

Fragmentation and Reassembly.

The internet identification field (ID) is used together with the source and destination address, and the protocol fields, to identify datagram fragments for reassembly.

The More Fragments flag bit (MF) is set if the datagram is not the last fragment. The Fragment Offset field identifies the fragment location, relative to the beginning of the original unfragmented datagram. Fragments are counted in units of 8 octets. The

fragmentation strategy is designed so that an unfragmented datagram has all zero fragmentation information (MF = 0, fragment offset = 0). If an internet datagram is fragmented, its data portion must be broken on 8 octet boundaries.

This format allows $2^{13} = 8192$ fragments of 8 octets each for a total of 65,536 octets. Note that this is consistent with the the datagram total length field (of course, the header is counted in the total length and not in the fragments).

When fragmentation occurs, some options are copied, but others remain with the first fragment only.

Every internet module must be able to forward a datagram of 68 octets without further fragmentation. This is because an internet header may be up to 60 octets, and the minimum fragment is 8 octets.

Every internet destination must be able to receive a datagram of 576 octets either in one piece or in fragments to be reassembled.

The fields which may be affected by fragmentation include:

- (1) options field
- (2) more fragments flag
- (3) fragment offset
- (4) internet header length field
- (5) total length field
- (6) header checksum

If the Don't Fragment flag (DF) bit is set, then internet fragmentation of this datagram is NOT permitted, although it may be discarded. This can be used to prohibit fragmentation in cases where the receiving host does not have sufficient resources to reassemble internet fragments.

One example of use of the Don't Fragment feature is to down line load a small host. A small host could have a boot strap program that accepts a datagram stores it in memory and then executes it.

The fragmentation and reassembly procedures are most easily described by examples. The following procedures are example implementations.

General notation in the following pseudo programs: " \leq " means "less than or equal", " \neq " means "not equal", " $=$ " means "equal", " \leftarrow " means "is set to". Also, " x to y " includes x and excludes y ; for example, "4 to 7" would include 4, 5, and 6 (but not 7).

An Example Fragmentation Procedure

The maximum sized datagram that can be transmitted through the next network is called the maximum transmission unit (MTU).

If the total length is less than or equal the maximum transmission unit then submit this datagram to the next step in datagram processing; otherwise cut the datagram into two fragments, the first fragment being the maximum size, and the second fragment being the rest of the datagram. The first fragment is submitted to the next step in datagram processing, while the second fragment is submitted to this procedure in case it is still too large.

Notation:

FO - Fragment Offset
 IHL - Internet Header Length
 DF - Don't Fragment flag
 MF - More Fragments flag
 TL - Total Length
 OFO - Old Fragment Offset
 OIHL - Old Internet Header Length
 OMF - Old More Fragments flag
 OTL - Old Total Length
 NFB - Number of Fragment Blocks
 MTU - Maximum Transmission Unit

Procedure:

IF $TL \leq MTU$ THEN Submit this datagram to the next step
 in datagram processing ELSE IF $DF = 1$ THEN discard the
 datagram ELSE

To produce the first fragment:

- (1) Copy the original internet header;
- (2) $OIHL \leftarrow IHL$; $OTL \leftarrow TL$; $OFO \leftarrow FO$; $OMF \leftarrow MF$;
- (3) $NFB \leftarrow (MTU - IHL * 4) / 8$;
- (4) Attach the first $NFB * 8$ data octets;
- (5) Correct the header:
 $MF \leftarrow 1$; $TL \leftarrow (IHL * 4) + (NFB * 8)$;
 Recompute Checksum;
- (6) Submit this fragment to the next step in
 datagram processing;

To produce the second fragment:

- (7) Selectively copy the internet header (some options
 are not copied, see option definitions);
- (8) Append the remaining data;
- (9) Correct the header:
 $IHL \leftarrow (((OIHL * 4) - (\text{length of options not copied})) + 3) / 4$;

$TL \leftarrow OTL - NFB * 8 - (OIHL - IHL) * 4$;

$FO \leftarrow OFO + NFB$; $MF \leftarrow OMF$; Recompute Checksum;

(10) Submit this fragment to the fragmentation test; DONE.

In the above procedure each fragment (except the last) was made the maximum allowable size. An alternative might produce less than the maximum size datagrams. For example, one could implement a fragmentation procedure that repeatedly divided large datagrams in half until the resulting fragments were less than the maximum transmission unit size.

An Example Reassembly Procedure

For each datagram the buffer identifier is computed as the concatenation of the source, destination, protocol, and identification fields. If this is a whole datagram (that is both the fragment offset and the more fragments fields are zero), then any reassembly resources associated with this buffer identifier are released and the datagram is forwarded to the next step in datagram processing.

If no other fragment with this buffer identifier is on hand then reassembly resources are allocated. The reassembly resources consist of a data buffer, a header buffer, a fragment block bit table, a total data length field, and a timer. The data from the fragment is placed in the data buffer according to its fragment offset and length, and bits are set in the fragment block bit table corresponding to the fragment blocks received.

If this is the first fragment (that is the fragment offset is zero) this header is placed in the header buffer. If this is the last fragment (that is the more fragments field is zero) the total data length is computed. If this fragment completes the datagram (tested by checking the bits set in the fragment block table), then the datagram is sent to the next step in datagram processing; otherwise the timer is set to the maximum of the current timer value and the value of the time to live field from this fragment; and the reassembly routine gives up control.

If the timer runs out, the all reassembly resources for this buffer identifier are released. The initial setting of the timer is a lower bound on the reassembly waiting time. This is because the waiting time will be increased if the Time to Live in the arriving fragment is greater than the current timer value but will not be decreased if it is less. The maximum this timer value could reach is the maximum time to live (approximately 4.25 minutes). The current recommendation for the initial timer setting is 15 seconds. This may be changed as experience with

Internet Protocol Specification

this protocol accumulates. Note that the choice of this parameter value is related to the buffer capacity available and the data rate of the transmission medium; that is, data rate times timer value equals buffer size (e.g., 10Kb/s X 15s = 150Kb).

Notation:

FO - Fragment Offset
 IHL - Internet Header Length
 MF - More Fragments flag
 TTL - Time To Live
 NFB - Number of Fragment Blocks
 TL - Total Length
 TDL - Total Data Length
 BUFID - Buffer Identifier
 RCVBT - Fragment Received Bit Table
 TLB - Timer Lower Bound

Procedure:

- (1) BUFID \leftarrow source|destination|protocol|identification;
- (2) IF FO = 0 AND MF = 0
- (3) THEN IF buffer with BUFID is allocated
- (4) THEN flush all reassembly for this BUFID;
- (5) Submit datagram to next step; DONE.
- (6) ELSE IF no buffer with BUFID is allocated
- (7) THEN allocate reassembly resources
 with BUFID;
 TIMER \leftarrow TLB; TDL \leftarrow 0;
- (8) put data from fragment into data buffer with
 BUFID from octet FO*8 to
 octet (TL-(IHL*4))+FO*8;
- (9) set RCVBT bits from FO
 to FO+((TL-(IHL*4)+7)/8);
- (10) IF MF = 0 THEN TDL \leftarrow TL-(IHL*4)+(FO*8)
- (11) IF FO = 0 THEN put header in header buffer
- (12) IF TDL \neq 0
- (13) AND all RCVBT bits from 0
 to (TDL+7)/8 are set
- (14) THEN TL \leftarrow TDL+(IHL*4)
- (15) Submit datagram to next step;
- (16) free all reassembly resources
 for this BUFID; DONE.
- (17) TIMER \leftarrow MAX(TIMER, TTL);
- (18) give up until next fragment or timer expires;
- (19) timer expires: flush all reassembly with this BUFID; DONE.

In the case that two or more fragments contain the same data

either identically or through a partial overlap, this procedure will use the more recently arrived copy in the data buffer and datagram delivered.

Identification

The choice of the Identifier for a datagram is based on the need to provide a way to uniquely identify the fragments of a particular datagram. The protocol module assembling fragments judges fragments to belong to the same datagram if they have the same source, destination, protocol, and Identifier. Thus, the sender must choose the Identifier to be unique for this source, destination pair and protocol for the time the datagram (or any fragment of it) could be alive in the internet.

It seems then that a sending protocol module needs to keep a table of Identifiers, one entry for each destination it has communicated with in the last maximum packet lifetime for the internet.

However, since the Identifier field allows 65,536 different values, some host may be able to simply use unique identifiers independent of destination.

It is appropriate for some higher level protocols to choose the identifier. For example, TCP protocol modules may retransmit an identical TCP segment, and the probability for correct reception would be enhanced if the retransmission carried the same identifier as the original transmission since fragments of either datagram could be used to construct a correct TCP segment.

Type of Service

The type of service (TOS) is for internet service quality selection. The type of service is specified along the abstract parameters precedence, delay, throughput, and reliability. These abstract parameters are to be mapped into the actual service parameters of the particular networks the datagram traverses.

Precedence. An independent measure of the importance of this datagram.

Delay. Prompt delivery is important for datagrams with this indication.

Throughput. High data rate is important for datagrams with this indication.

Internet Protocol
Specification

Reliability. A higher level of effort to ensure delivery is important for datagrams with this indication.

For example, the ARPANET has a priority bit, and a choice between "standard" messages (type 0) and "uncontrolled" messages (type 3), (the choice between single packet and multipacket messages can also be considered a service parameter). The uncontrolled messages tend to be less reliably delivered and suffer less delay. Suppose an internet datagram is to be sent through the ARPANET. Let the internet type of service be given as:

Precedence: 5
Delay: 0
Throughput: 1
Reliability: 1

In this example, the mapping of these parameters to those available for the ARPANET would be to set the ARPANET priority bit on since the Internet precedence is in the upper half of its range, to select standard messages since the throughput and reliability requirements are indicated and delay is not. More details are given on service mappings in "Service Mappings" [8].

Time to Live

The time to live is set by the sender to the maximum time the datagram is allowed to be in the internet system. If the datagram is in the internet system longer than the time to live, then the datagram must be destroyed.

This field must be decreased at each point that the internet header is processed to reflect the time spent processing the datagram. Even if no local information is available on the time actually spent, the field must be decremented by 1. The time is measured in units of seconds (i.e. the value 1 means one second). Thus, the maximum time to live is 255 seconds or 4.25 minutes. Since every module that processes a datagram must decrease the TTL by at least one even if it process the datagram in less than a second, the TTL must be thought of only as an upper bound on the time a datagram may exist. The intention is to cause undeliverable datagrams to be discarded, and to bound the maximum datagram lifetime.

Some higher level reliable connection protocols are based on assumptions that old duplicate datagrams will not arrive after a certain time elapses. The TTL is a way for such protocols to have an assurance that their assumption is met.

Options

The options are optional in each datagram, but required in implementations. That is, the presence or absence of an option is the choice of the sender, but each internet module must be able to parse every option. There can be several options present in the option field.

The options might not end on a 32-bit boundary. The internet header must be filled out with octets of zeros. The first of these would be interpreted as the end-of-options option, and the remainder as internet header padding.

Every internet module must be able to act on every option. The Security Option is required if classified, restricted, or compartmented traffic is to be passed.

Checksum

The internet header checksum is recomputed if the internet header is changed. For example, a reduction of the time to live, additions or changes to internet options, or due to fragmentation. This checksum at the internet level is intended to protect the internet header fields from transmission errors.

There are some applications where a few data bit errors are acceptable while retransmission delays are not. If the internet protocol enforced data correctness such applications could not be supported.

Errors

Internet protocol errors may be reported via the ICMP messages [3].

3.3. Interfaces

The functional description of user interfaces to the IP is, at best, fictional, since every operating system will have different facilities. Consequently, we must warn readers that different IP implementations may have different user interfaces. However, all IPs must provide a certain minimum set of services to guarantee that all IP implementations can support the same protocol hierarchy. This section specifies the functional interfaces required of all IP implementations.

Internet protocol interfaces on one side to the local network and on the other side to either a higher level protocol or an application program. In the following, the higher level protocol or application

Internet Protocol Specification

program (or even a gateway program) will be called the "user" since it is using the internet module. Since internet protocol is a datagram protocol, there is minimal memory or state maintained between datagram transmissions, and each call on the internet protocol module by the user supplies all information necessary for the IP to perform the service requested.

An Example Upper Level Interface

The following two example calls satisfy the requirements for the user to internet protocol module communication ("=>" means returns):

SEND (src, dst, prot, TOS, TTL, BufPTR, len, Id, DF, opt => result)

where:

src = source address
 dst = destination address
 prot = protocol
 TOS = type of service
 TTL = time to live
 BufPTR = buffer pointer
 len = length of buffer
 Id = Identifier
 DF = Don't Fragment
 opt = option data
 result = response
 OK = datagram sent ok
 Error = error in arguments or local network error

Note that the precedence is included in the TOS and the security/compartments is passed as an option.

RECV (BufPTR, prot, => result, src, dst, TOS, len, opt)

where:

BufPTR = buffer pointer
 prot = protocol
 result = response
 OK = datagram received ok
 Error = error in arguments
 len = length of buffer
 src = source address
 dst = destination address
 TOS = type of service
 opt = option data

When the user sends a datagram, it executes the SEND call supplying all the arguments. The internet protocol module, on receiving this call, checks the arguments and prepares and sends the message. If the arguments are good and the datagram is accepted by the local network, the call returns successfully. If either the arguments are bad, or the datagram is not accepted by the local network, the call returns unsuccessfully. On unsuccessful returns, a reasonable report must be made as to the cause of the problem, but the details of such reports are up to individual implementations.

When a datagram arrives at the internet protocol module from the local network, either there is a pending RECV call from the user addressed or there is not. In the first case, the pending call is satisfied by passing the information from the datagram to the user. In the second case, the user addressed is notified of a pending datagram. If the user addressed does not exist, an ICMP error message is returned to the sender, and the data is discarded.

The notification of a user may be via a pseudo interrupt or similar mechanism, as appropriate in the particular operating system environment of the implementation.

A user's RECV call may then either be immediately satisfied by a pending datagram, or the call may be pending until a datagram arrives.

The source address is included in the send call in case the sending host has several addresses (multiple physical connections or logical addresses). The internet module must check to see that the source address is one of the legal address for this host.

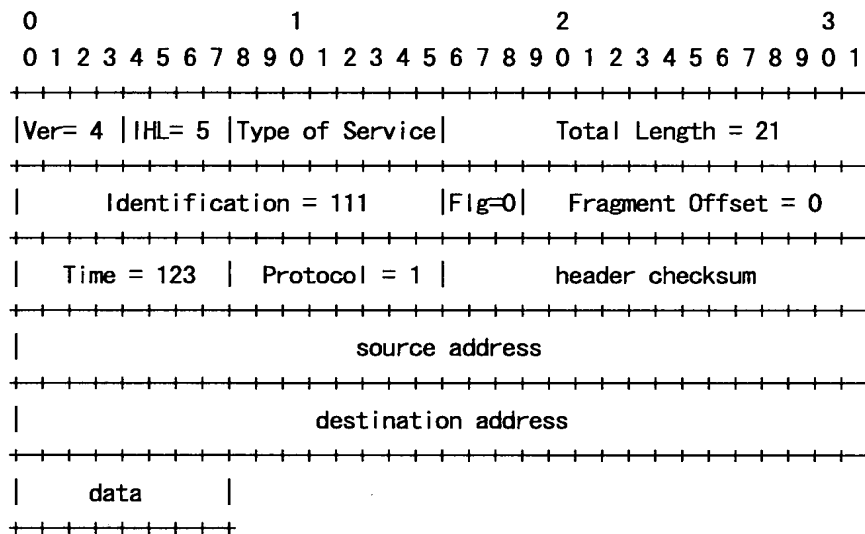
An implementation may also allow or require a call to the internet module to indicate interest in or reserve exclusive use of a class of datagrams (e.g., all those with a certain value in the protocol field).

This section functionally characterizes a USER/IP interface. The notation used is similar to most procedure of function calls in high level languages, but this usage is not meant to rule out trap type service calls (e.g., SVCs, UUOs, EMTs), or any other form of interprocess communication.

APPENDIX A: Examples & Scenarios

Example 1:

This is an example of the minimal data carrying internet datagram:



Example Internet Datagram

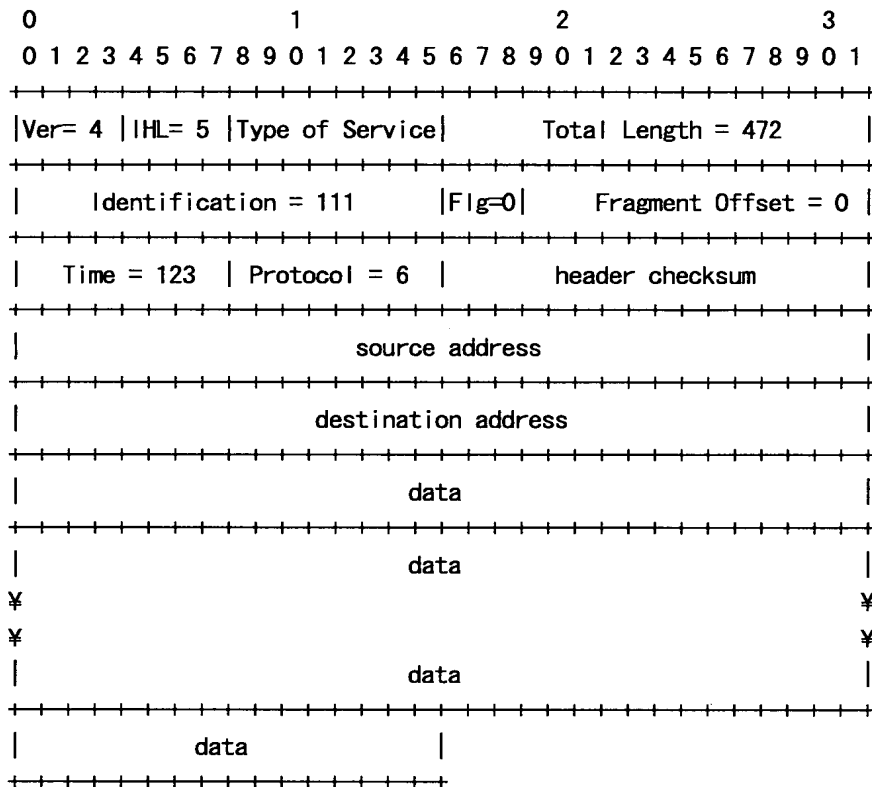
Figure 5.

Note that each tick mark represents one bit position.

This is a internet datagram in version 4 of internet protocol; the internet header consists of five 32 bit words, and the total length of the datagram is 21 octets. This datagram is a complete datagram (not a fragment).

Example 2:

In this example, we show first a moderate size internet datagram (452 data octets), then two internet fragments that might result from the fragmentation of this datagram if the maximum sized transmission allowed were 280 octets.



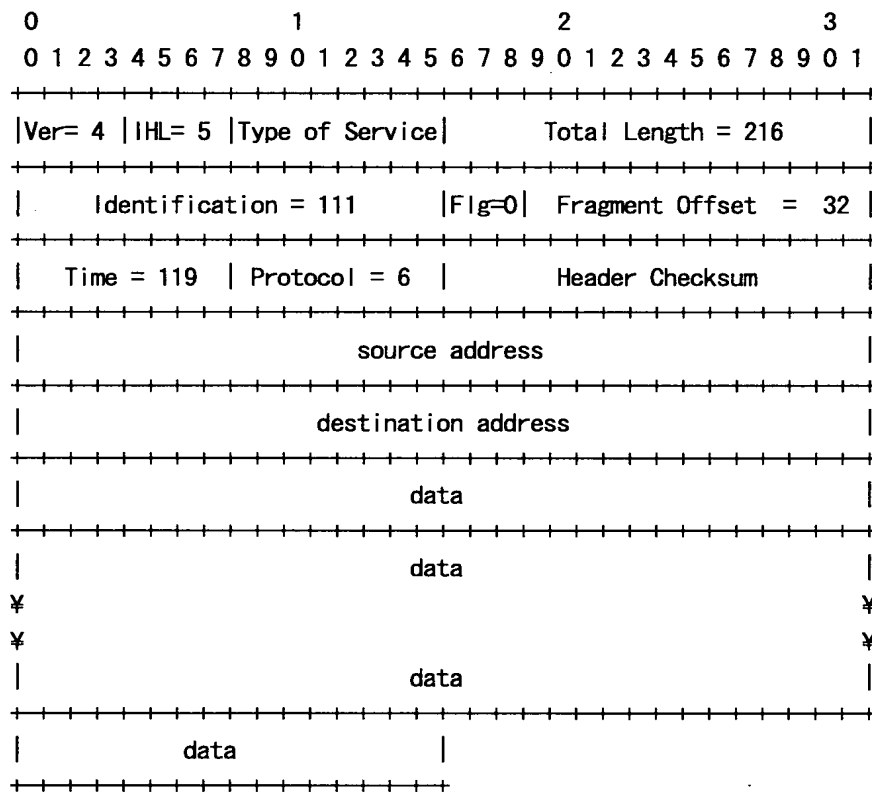
Example Internet Datagram

Figure 6.

0										1										2										3									
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
Ver= 4 IHL= 5 Type of Service										Total Length = 276																													
Identification = 111										Flg=1										Fragment Offset = 0																			
Time = 119										Protocol = 6										Header Checksum																			
										source address																													
										destination address																													
										data																													
										data																													
										data																													
										data																													

Figure 7.

And the second fragment.



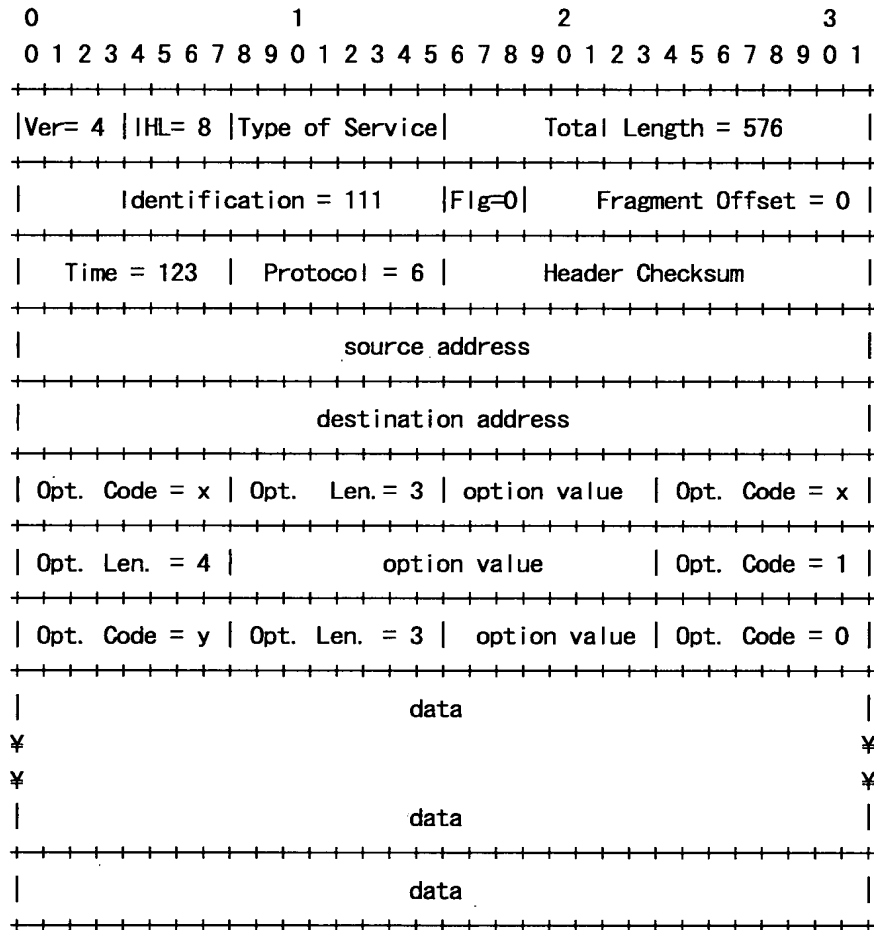
Example Internet Fragment

Figure 8.

Internet Protocol

Example 3:

Here, we show an example of a datagram containing options:

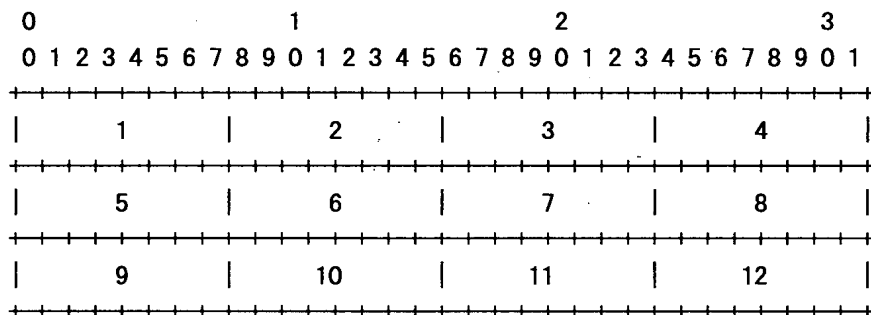


Example Internet Datagram

Figure 9.

APPENDIX B: Data Transmission Order

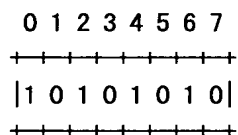
The order of transmission of the header and data described in this document is resolved to the octet level. Whenever a diagram shows a group of octets, the order of transmission of those octets is the normal order in which they are read in English. For example, in the following diagram the octets are transmitted in the order they are numbered.



Transmission Order of Bytes

Figure 10.

Whenever an octet represents a numeric quantity the left most bit in the diagram is the high order or most significant bit. That is, the bit labeled 0 is the most significant bit. For example, the following diagram represents the value 170 (decimal).



Significance of Bits

Figure 11.

Similarly, whenever a multi-octet field represents a numeric quantity the left most bit of the whole field is the most significant bit. When a multi-octet quantity is transmitted the most significant octet is transmitted first.

GLOSSARY

1822

BBN Report 1822, "The Specification of the Interconnection of a Host and an IMP". The specification of interface between a host and the ARPANET.

ARPANET leader

The control information on an ARPANET message at the host-IMP interface.

ARPANET message

The unit of transmission between a host and an IMP in the ARPANET. The maximum size is about 1012 octets (8096 bits).

ARPANET packet

A unit of transmission used internally in the ARPANET between IMPs. The maximum size is about 126 octets (1008 bits).

Destination

The destination address, an internet header field.

DF

The Don't Fragment bit carried in the flags field.

Flags

An internet header field carrying various control flags.

Fragment Offset

This internet header field indicates where in the internet datagram a fragment belongs.

GGP

Gateway to Gateway Protocol, the protocol used primarily between gateways to control routing and other gateway functions.

header

Control information at the beginning of a message, segment, datagram, packet or block of data.

ICMP

Internet Control Message Protocol, implemented in the internet module, the ICMP is used from gateways to hosts and between hosts to report errors and make routing suggestions.

Internet Protocol
Glossary

Identification

An internet header field carrying the identifying value assigned by the sender to aid in assembling the fragments of a datagram.

IHL

The internet header field Internet Header Length is the length of the internet header measured in 32 bit words.

IMP

The Interface Message Processor, the packet switch of the ARPANET.

Internet Address

A four octet (32 bit) source or destination address consisting of a Network field and a Local Address field.

internet datagram

The unit of data exchanged between a pair of internet modules (includes the internet header).

internet fragment

A portion of the data of an internet datagram with an internet header.

Local Address

The address of a host within a network. The actual mapping of an internet local address on to the host addresses in a network is quite general, allowing for many to one mappings.

MF

The More-Fragments Flag carried in the internet header flags field.

module

An implementation, usually in software, of a protocol or other procedure.

more-fragments flag

A flag indicating whether or not this internet datagram contains the end of an internet datagram, carried in the internet header Flags field.

NFB

The Number of Fragment Blocks in a the data portion of an internet fragment. That is, the length of a portion of data measured in 8 octet units.

octet

An eight bit byte.

Options

The internet header Options field may contain several options, and each option may be several octets in length.

Padding

The internet header Padding field is used to ensure that the data begins on 32 bit word boundary. The padding is zero.

Protocol

In this document, the next higher level protocol identifier, an internet header field.

Rest

The local address portion of an Internet Address.

Source

The source address, an internet header field.

TCP

Transmission Control Protocol: A host-to-host protocol for reliable communication in internet environments.

TCP Segment

The unit of data exchanged between TCP modules (including the TCP header).

TFTP

Trivial File Transfer Protocol: A simple file transfer protocol built on UDP.

Time to Live

An internet header field which indicates the upper bound on how long this internet datagram may exist.

TOS

Type of Service

Total Length

The internet header field Total Length is the length of the datagram in octets including internet header and data.

TTL

Time to Live

Internet Protocol
Glossary

Type of Service

An internet header field which indicates the type (or quality) of service for this internet datagram.

UDP

User Datagram Protocol: A user level protocol for transaction oriented applications.

User

The user of the internet protocol. This may be a higher level protocol module, an application program, or a gateway program.

Version

The Version field indicates the format of the internet header.

REFERENCES

- [1] Cerf, V., "The Catenet Model for Internetworking," Information Processing Techniques Office, Defense Advanced Research Projects Agency, IEN 48, July 1978.
- [2] Bolt Beranek and Newman, "Specification for the Interconnection of a Host and an IMP," BBN Technical Report 1822, Revised May 1978.
- [3] Postel, J., "Internet Control Message Protocol - DARPA Internet Program Protocol Specification," RFC 792, USC/Information Sciences Institute, September 1981.
- [4] Shoch, J., "Inter-Network Naming, Addressing, and Routing," COMPCON, IEEE Computer Society, Fall 1978.
- [5] Postel, J., "Address Mappings," RFC 796, USC/Information Sciences Institute, September 1981.
- [6] Shoch, J., "Packet Fragmentation in Inter-Network Protocols," Computer Networks, v. 3, n. 1, February 1979.
- [7] Strazisar, V., "How to Build a Gateway", IEN 109, Bolt Beranek and Newman, August 1979.
- [8] Postel, J., "Service Mappings," RFC 795, USC/Information Sciences Institute, September 1981.
- [9] Postel, J., "Assigned Numbers," RFC 790, USC/Information Sciences Institute, September 1981.

IETF Mobile IP Working Group
Internet-Draft
Expires: December 29, 2003

D. Johnson
Rice University
C. Perkins
Nokia Research Center
J. Arkko
Ericsson
June 30, 2003

Mobility Support in IPv6
draft-ietf-mobileip-ipv6-24.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on December 29, 2003.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This document specifies a protocol which allows nodes to remain reachable while moving around in the IPv6 Internet. Each mobile node is always identified by its home address, regardless of its current point of attachment to the Internet. While situated away from its home, a mobile node is also associated with a care-of address, which provides information about the mobile node's current location. IPv6 packets addressed to a mobile node's home address are transparently routed to its care-of address. The protocol enables IPv6 nodes to

cache the binding of a mobile node's home address with its care-of address, and to then send any packets destined for the mobile node directly to it at this care-of address. To support this operation, Mobile IPv6 defines a new IPv6 protocol and a new destination option. All IPv6 nodes, whether mobile or stationary can communicate with mobile nodes.

Table of Contents

1.	Introduction	6
2.	Comparison with Mobile IP for IPv4	8
3.	Terminology	9
3.1	General Terms	9
3.2	Mobile IPv6 Terms	11
4.	Overview of Mobile IPv6	15
4.1	Basic Operation	15
4.2	New IPv6 Protocol	17
4.3	New IPv6 Destination Option	18
4.4	New IPv6 ICMP Messages	18
4.5	Conceptual Data Structure Terminology	18
4.6	Site-Local Addressability	19
5.	Overview of Mobile IPv6 Security	20
5.1	Binding Updates to Home Agents	20
5.2	Binding Updates to Correspondent Nodes	21
5.2.1	Node Keys	22
5.2.2	Nonces	22
5.2.3	Cookies and Tokens	23
5.2.4	Cryptographic Functions	23
5.2.5	Return Routability Procedure	23
5.2.6	Authorizing Binding Management Messages	28
5.2.7	Updating Node Keys and Nonces	30
5.2.8	Preventing Replay Attacks	31
5.3	Dynamic Home Agent Address Discovery	31
5.4	Mobile Prefix Discovery	31
5.5	Payload Packets	32
6.	New IPv6 Protocol, Message Types, and Destination Option	33
6.1	Mobility Header	33
6.1.1	Format	33
6.1.2	Binding Refresh Request Message	35
6.1.3	Home Test Init Message	36
6.1.4	Care-of Test Init Message	37
6.1.5	Home Test Message	38
6.1.6	Care-of Test Message	39
6.1.7	Binding Update Message	41
6.1.8	Binding Acknowledgement Message	43
6.1.9	Binding Error Message	46
6.2	Mobility Options	47
6.2.1	Format	48

6.2.2	Pad1	48
6.2.3	PadN	49
6.2.4	Binding Refresh Advice	49
6.2.5	Alternate Care-of Address	50
6.2.6	Nonce Indices	50
6.2.7	Binding Authorization Data	51
6.3	Home Address Option	52
6.4	Type 2 Routing Header	54
6.4.1	Format	54
6.5	ICMP Home Agent Address Discovery Request Message	56
6.6	ICMP Home Agent Address Discovery Reply Message	57
6.7	ICMP Mobile Prefix Solicitation Message Format	58
6.8	ICMP Mobile Prefix Advertisement Message Format	60
7.	Modifications to IPv6 Neighbor Discovery	63
7.1	Modified Router Advertisement Message Format	63
7.2	Modified Prefix Information Option Format	63
7.3	New Advertisement Interval Option Format	65
7.4	New Home Agent Information Option Format	66
7.5	Changes to Sending Router Advertisements	68
8.	Requirements for Types of IPv6 Nodes	70
8.1	All IPv6 Nodes	70
8.2	IPv6 Nodes with Support for Route Optimization	70
8.3	All IPv6 Routers	72
8.4	IPv6 Home Agents	72
8.5	IPv6 Mobile Nodes	74
9.	Correspondent Node Operation	76
9.1	Conceptual Data Structures	76
9.2	Processing Mobility Headers	77
9.3	Packet Processing	77
9.3.1	Receiving Packets with Home Address Option	77
9.3.2	Sending Packets to a Mobile Node	78
9.3.3	Sending Binding Error Messages	80
9.3.4	Receiving ICMP Error Messages	80
9.4	Return Routability Procedure	81
9.4.1	Receiving Home Test Init Messages	81
9.4.2	Receiving Care-of Test Init Messages	81
9.4.3	Sending Home Test Messages	82
9.4.4	Sending Care-of Test Messages	82
9.5	Processing Bindings	82
9.5.1	Receiving Binding Updates	82
9.5.2	Requests to Cache a Binding	85
9.5.3	Requests to Delete a Binding	85
9.5.4	Sending Binding Acknowledgements	86
9.5.5	Sending Binding Refresh Requests	87
9.6	Cache Replacement Policy	87
10.	Home Agent Operation	89
10.1	Conceptual Data Structures	89
10.2	Processing Mobility Headers	90

10.3	Processing Bindings	90
10.3.1	Primary Care-of Address Registration	90
10.3.2	Primary Care-of Address De-Registration	94
10.4	Packet Processing	95
10.4.1	Intercepting Packets for a Mobile Node	95
10.4.2	Processing Intercepted Packets	97
10.4.3	Multicast Membership Control	98
10.4.4	Stateful Address Autoconfiguration	99
10.4.5	Handling Reverse Tunneled Packets	99
10.4.6	Protecting Return Routability Packets	100
10.5	Dynamic Home Agent Address Discovery	101
10.5.1	Receiving Router Advertisement Messages	101
10.6	Sending Prefix Information to the Mobile Node	103
10.6.1	List of Home Network Prefixes	103
10.6.2	Scheduling Prefix Deliveries	104
10.6.3	Sending Advertisements	106
10.6.4	Lifetimes for Changed Prefixes	107
11.	Mobile Node Operation	108
11.1	Conceptual Data Structures	108
11.2	Processing Mobility Headers	109
11.3	Packet Processing	110
11.3.1	Sending Packets While Away from Home	110
11.3.2	Interaction with Outbound IPsec Processing	113
11.3.3	Receiving Packets While Away from Home	115
11.3.4	Routing Multicast Packets	116
11.3.5	Receiving ICMP Error Messages	118
11.3.6	Receiving Binding Error Messages	118
11.4	Home Agent and Prefix Management	119
11.4.1	Dynamic Home Agent Address Discovery	119
11.4.2	Sending Mobile Prefix Solicitations	120
11.4.3	Receiving Mobile Prefix Advertisements	121
11.5	Movement	122
11.5.1	Movement Detection	122
11.5.2	Forming New Care-of Addresses	124
11.5.3	Using Multiple Care-of Addresses	125
11.5.4	Returning Home	126
11.6	Return Routability Procedure	128
11.6.1	Sending Test Init Messages	128
11.6.2	Receiving Test Messages	129
11.6.3	Protecting Return Routability Packets	130
11.7	Processing Bindings	130
11.7.1	Sending Binding Updates to the Home Agent	131
11.7.2	Correspondent Registration	133
11.7.3	Receiving Binding Acknowledgements	136
11.7.4	Receiving Binding Refresh Requests	138
11.8	Retransmissions and Rate Limiting	139
12.	Protocol Constants	141
13.	Protocol Configuration Variables	142

14.	IANA Considerations	143
15.	Security Considerations	145
15.1	Threats	145
15.2	Features	147
15.3	Binding Updates to Home Agent	148
15.4	Binding Updates to Correspondent Nodes	151
15.4.1	Overview	151
15.4.2	Achieved Security Properties	152
15.4.3	Comparison to Regular IPv6 Communications	153
15.4.4	Replay Attacks	155
15.4.5	Denial-of-Service Attacks	155
15.4.6	Key Lengths	156
15.5	Dynamic Home Agent Address Discovery	157
15.6	Mobile Prefix Discovery	157
15.7	Tunneling via the Home Agent	158
15.8	Home Address Option	158
15.9	Type 2 Routing Header	159
16.	Contributors	161
17.	Acknowledgements	162
	Normative References	163
	Informative References	165
	Authors' Addresses	166
A.	Changes from Previous Version of the Draft	167
B.	Future Extensions	168
B.1	Piggybacking	168
B.2	Triangular Routing	168
B.3	New Authorization Methods	168
B.4	Dynamically Generated Home Addresses	168
B.5	Remote Home Address Configuration	168
B.6	Neighbor Discovery Extensions	169
	Intellectual Property and Copyright Statements	171

1. Introduction

This document specifies a protocol which allows nodes to remain reachable while moving around in the IPv6 Internet. Without specific support for mobility in IPv6 [11], packets destined to a mobile node would not be able to reach it while the mobile node is away from its home link. In order to continue communication in spite of its movement, a mobile node could change its IP address each time it moves to a new link, but the mobile node would then not be able to maintain transport and higher-layer connections when it changes location. Mobility support in IPv6 is particularly important, as mobile computers are likely to account for a majority or at least a substantial fraction of the population of the Internet during the lifetime of IPv6.

The protocol defined in this document, known as Mobile IPv6, allows a mobile node to move from one link to another without changing the mobile node's "home address". Packets may be routed to the mobile node using this address regardless of the mobile node's current point of attachment to the Internet. The mobile node may also continue to communicate with other nodes (stationary or mobile) after moving to a new link. The movement of a mobile node away from its home link is thus transparent to transport and higher-layer protocols and applications.

The Mobile IPv6 protocol is just as suitable for mobility across homogeneous media as for mobility across heterogeneous media. For example, Mobile IPv6 facilitates node movement from one Ethernet segment to another as well as it facilitates node movement from an Ethernet segment to a wireless LAN cell, with the mobile node's IP address remaining unchanged in spite of such movement.

One can think of the Mobile IPv6 protocol as solving the network-layer mobility management problem. Some mobility management applications — for example, handover among wireless transceivers, each of which covers only a very small geographic area — have been solved using link-layer techniques. For example, in many current wireless LAN products, link-layer mobility mechanisms allow a "handover" of a mobile node from one cell to another, re-establishing link-layer connectivity to the node in each new location.

Mobile IPv6 does not attempt to solve all general problems related to the use of mobile computers or wireless networks. In particular, this protocol does not attempt to solve:

- o Handling links with unidirectional connectivity or partial reachability, such as the hidden terminal problem where a host is hidden from only some of the routers on the link.

- o Access control on a link being visited by a mobile node.
- o Local or hierarchical forms of mobility management (similar to many current link-layer mobility management solutions).
- o Assistance for adaptive applications
- o Mobile routers
- o Service Discovery
- o Distinguishing between packets lost due to bit errors vs. network congestion

2. Comparison with Mobile IP for IPv4

The design of Mobile IP support in IPv6 (Mobile IPv6) benefits both from the experiences gained from the development of Mobile IP support in IPv4 (Mobile IPv4) [22, 23, 24], and from the opportunities provided by IPv6. Mobile IPv6 thus shares many features with Mobile IPv4, but is integrated into IPv6 and offers many other improvements. This section summarizes the major differences between Mobile IPv4 and Mobile IPv6:

- o There is no need to deploy special routers as "foreign agents", as in Mobile IPv4. Mobile IPv6 operates in any location without any special support required from the local router.
- o Support for route optimization is a fundamental part of the protocol, rather than a nonstandard set of extensions.
- o Mobile IPv6 route optimization can operate securely even without pre-arranged security associations. It is expected that route optimization can be deployed on a global scale between all mobile nodes and correspondent nodes.
- o Support is also integrated into Mobile IPv6 for allowing route optimization to coexist efficiently with routers that perform "ingress filtering" [26].
- o The IPv6 Neighbor Unreachability Detection assures symmetric reachability between the mobile node and its default router in the current location.
- o Most packets sent to a mobile node while away from home in Mobile IPv6 are sent using an IPv6 routing header rather than IP encapsulation, reducing the amount of resulting overhead compared to Mobile IPv4.
- o Mobile IPv6 is decoupled from any particular link layer, as it uses IPv6 Neighbor Discovery [12] instead of ARP. This also improves the robustness of the protocol.
- o The use of IPv6 encapsulation (and the routing header) removes the need in Mobile IPv6 to manage "tunnel soft state".
- o The dynamic home agent address discovery mechanism in Mobile IPv6 returns a single reply to the mobile node. The directed broadcast approach used in IPv4 returns separate replies from each home agent.

3. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [2].

3.1 General Terms

IP

Internet Protocol Version 6 (IPv6).

node

A device that implements IP.

router

A node that forwards IP packets not explicitly addressed to itself.

unicast routable address

An identifier for a single interface such that a packet sent to it from another IPv6 subnet is delivered to the interface identified by that address. Accordingly, a unicast routable address must have either a global or site-local scope (but not link-local).

host

Any node that is not a router.

link

A communication facility or medium over which nodes can communicate at the link layer, such as an Ethernet (simple or bridged). A link is the layer immediately below IP.

interface

A node's attachment to a link.

subnet prefix

A bit string that consists of some number of initial bits of an IP address.

interface identifier

A number used to identify a node's interface on a link. The interface identifier is the remaining low-order bits in the node's IP address after the subnet prefix.

link-layer address

A link-layer identifier for an interface, such as IEEE 802 addresses on Ethernet links.

packet

An IP header plus payload.

security association

An IPsec security association is a cooperative relationship formed by the sharing of cryptographic keying material and associated context. Security associations are simplex. That is, two security associations are needed to protect bidirectional traffic between two nodes, one for each direction.

security policy database

A database that specifies what security services are to be offered to IP packets and in what fashion.

destination option

Destination options are carried by the IPv6 Destination Options extension header. Destination options include optional information that need be examined only by the IPv6 node given as the destination address in the IPv6 header, not by routers in between. Mobile IPv6 defines one new destination option, the Home Address destination option (see Section 6.3).

routing header

A routing header may be present as an IPv6 header extension, and indicates that the payload has to be delivered to a destination IPv6 address in some way that is different from what would be carried out by standard Internet routing. In this document, use of the term "routing header" typically refers to use of a type 2 routing header, as specified in Section 6.4.

'|' (concatenation)

Some formulas in this specification use the symbol '|' indicate bitwise concatenation, as in $A | B$. This concatenation requires that all of the octets of the datum A appear first in the result, followed by all of the octets of the datum B.

First (size, input)

Some formulas in this specification use a functional form "First (size, input)" to indicate truncation of the "input" data so that only the first "size" bits remain to be used.

3.2 Mobile IPv6 Terms

home address

A unicast routable address assigned to a mobile node, used as the permanent address of the mobile node. This address is within the mobile node's home link. Standard IP routing mechanisms will deliver packets destined for a mobile node's home address to its home link. Mobile nodes can have multiple home addresses, for instance when there are multiple home prefixes on the home link.

home subnet prefix

The IP subnet prefix corresponding to a mobile node's home address.

home link

The link on which a mobile node's home subnet prefix is defined.

mobile node

A node that can change its point of attachment from one link to another, while still being reachable via its home address.

movement

A change in a mobile node's point of attachment to the Internet such that it is no longer connected to the same link as it was previously. If a mobile node is not currently attached to its home link, the mobile node is said to be "away from home".

L2 handover

A process by which the mobile node changes from one link-layer connection to another. For example, a change of wireless access point is an L2 handover.

L3 handover

Subsequent to an L2 handover, a mobile node detects a change in an on-link subnet prefix that would require a change in the primary care-of address. For example, a change of access router subsequent to a change of wireless access point typically results in an L3 handover.

correspondent node

A peer node with which a mobile node is communicating. The correspondent node may be either mobile or stationary.

foreign subnet prefix

Any IP subnet prefix other than the mobile node's home subnet prefix.

foreign link

Any link other than the mobile node's home link.

care-of address

A unicast routable address associated with a mobile node while visiting a foreign link; the subnet prefix of this IP address is a foreign subnet prefix. Among the multiple care-of addresses that a mobile node may have at any given time (e.g., with different subnet prefixes), the one registered with the mobile node's home agent for a given home address is called its "primary" care-of address.

home agent

A router on a mobile node's home link with which the mobile node has registered its current care-of address. While the mobile node is away from home, the home agent intercepts packets on the home link destined to the mobile node's home address, encapsulates them, and tunnels them to the mobile node's registered care-of address.

binding

The association of the home address of a mobile node with a care-of address for that mobile node, along with the remaining lifetime of that association.

registration

The process during which a mobile node sends a Binding Update to its home agent or a correspondent node, causing a binding for the mobile node to be registered.

mobility message

A message containing a Mobility Header (see Section 6.1).

binding authorization

Correspondent registration needs to be authorized to allow the recipient to believe that the sender has the right to specify a new binding.

return routability procedure

The return routability procedure authorizes registrations by the use of a cryptographic token exchange.

correspondent registration

A return routability procedure followed by a registration, run between the mobile node and a correspondent node.

home registration

A registration between the mobile node and its home agent, authorized by the use of IPsec.

nonce

Nonces are random numbers used internally by the correspondent node in the creation of keygen tokens related to the return routability procedure. The nonces are not specific to a mobile node, and are kept secret within the correspondent node.

nonce index

A nonce index is used to indicate which nonces have been used when creating keygen token values, without revealing the nonces

themselves.

cookie

A cookie is a random number used by a mobile nodes to prevent spoofing by a bogus correspondent node in the return routability procedure.

care-of init cookie

A cookie sent to the correspondent node in the Care-of Test Init message, to be returned in the Care-of Test message.

home init cookie

A cookie sent to the correspondent node in the Home Test Init message, to be returned in the Home Test message.

keygen token

A keygen token is a number supplied by a correspondent node in the return routability procedure to enable the mobile node to compute the necessary binding management key for authorizing a Binding Update.

care-of keygen token

A keygen token sent by the correspondent node in the Care-of Test message.

home keygen token

A keygen token sent by the correspondent node in the Home Test message.

binding management key (Kbm)

A binding management key (Kbm) is a key used for authorizing a binding cache management message (e.g., Binding Update or Binding Acknowledgement). Return routability provides a way to create a binding management key.

4. Overview of Mobile IPv6

4.1 Basic Operation

A mobile node is always expected to be addressable at its home address, whether it is currently attached to its home link or is away from home. The "home address" is an IP address assigned to the mobile node within its home subnet prefix on its home link. While a mobile node is at home, packets addressed to its home address are routed to the mobile node's home link, using conventional Internet routing mechanisms.

While a mobile node is attached to some foreign link away from home, it is also addressable at one or more care-of addresses. A care-of address is an IP address associated with a mobile node that has the subnet prefix of a particular foreign link. The mobile node can acquire its care-of address through conventional IPv6 mechanisms, such as stateless or stateful auto-configuration. As long as the mobile node stays in this location, packets addressed to this care-of address will be routed to the mobile node. The mobile node may also accept packets from several care-of addresses, such as when it is moving but still reachable at the previous link.

The association between a mobile node's home address and care-of address is known as a "binding" for the mobile node. While away from home, a mobile node registers its primary care-of address with a router on its home link, requesting this router to function as the "home agent" for the mobile node. The mobile node performs this binding registration by sending a "Binding Update" message to the home agent. The home agent replies to the mobile node by returning a "Binding Acknowledgement" message. The operation of the mobile node is specified in Section 11, and the operation of the home agent is specified in Section 10.

Any node communicating with a mobile node is referred to in this document as a "correspondent node" of the mobile node, and may itself be either a stationary node or a mobile node. Mobile nodes can provide information about their current location to correspondent nodes. This happens through the correspondent registration. As a part of this procedure, a return routability test is performed in order to authorize the establishment of the binding. The operation of the correspondent node is specified in Section 9.

There are two possible modes for communications between the mobile node and a correspondent node. The first mode, bidirectional tunneling, does not require Mobile IPv6 support from the correspondent node and is available even if the mobile node has not registered its current binding with the correspondent node. Packets

from the correspondent node are routed to the home agent and then tunneled to the mobile node. Packets to the correspondent node are tunneled from the mobile node to the home agent ("reverse tunneled") and then routed normally from the home network to the correspondent node. In this mode, the home agent uses proxy Neighbor Discovery to intercept any IPv6 packets addressed to the mobile node's home address (or home addresses) on the home link. Each intercepted packet is tunneled to the mobile node's primary care-of address. This tunneling is performed using IPv6 encapsulation [15].

The second mode, "route optimization", requires the mobile node to register its current binding at the correspondent node. Packets from the correspondent node can be routed directly to the care-of address of the mobile node. When sending a packet to any IPv6 destination, the correspondent node checks its cached bindings for an entry for the packet's destination address. If a cached binding for this destination address is found, the node uses a new type of IPv6 routing header [11] (see Section 6.4) to route the packet to the mobile node by way of the care-of address indicated in this binding.

Routing packets directly to the mobile node's care-of address allows the shortest communications path to be used. It also eliminates congestion at the mobile node's home agent and home link. In addition, the impact of any possible failure of the home agent or networks on the path to or from it is reduced.

When routing packets directly to the mobile node, the correspondent node sets the Destination Address in the IPv6 header to the care-of address of the mobile node. A new type of IPv6 routing header (see Section 6.4) is also added to the packet to carry the desired home address. Similarly, the mobile node sets the Source Address in the packet's IPv6 header to its current care-of addresses. The mobile node adds a new IPv6 "Home Address" destination option (see Section 6.3) to carry its home address. The inclusion of home addresses in these packets makes the use of the care-of address transparent above the network layer (e.g., at the transport layer).

Mobile IPv6 also provides support for multiple home agents, and a limited support for the reconfiguration of the home network. In these cases, the mobile node may not know the IP address of its own home agent, and even the home subnet prefixes may change over time. A mechanism, known as "dynamic home agent address discovery" allows a mobile node to dynamically discover the IP address of a home agent on its home link, even when the mobile node is away from home. Mobile nodes can also learn new information about home subnet prefixes through the "mobile prefix discovery" mechanism. These mechanisms are described starting from Section 6.5.

4.2 New IPv6 Protocol

Mobile IPv6 defines a new IPv6 protocol, using the Mobility Header (see Section 6.1). This Header is used to carry the following messages:

Home Test Init

Home Test

Care-of Test Init

Care-of Test

These four messages are used to perform the return routability procedure from the mobile node to a correspondent node. This ensures authorization of subsequent Binding Updates, as described in Section 5.2.5.

Binding Update

A Binding Update is used by a mobile node to notify a correspondent node or the mobile node's home agent of its current binding. The Binding Update sent to the mobile node's home agent to register its primary care-of address is marked as a "home registration".

Binding Acknowledgement

A Binding Acknowledgement is used to acknowledge receipt of a Binding Update, if an acknowledgement was requested in the Binding Update, the binding update was sent to a home agent, or an error occurred.

Binding Refresh Request

A Binding Refresh Request is used by a correspondent node to request a mobile node to re-establish its binding with the correspondent node. This message is typically used when the cached binding is in active use but the binding's lifetime is close to expiration. The correspondent node may use, for instance, recent traffic and open transport layer connections as an indication of active use.

Binding Error

The Binding Error is used by the correspondent node to signal an error related to mobility, such as an inappropriate attempt to use

the Home Address destination option without an existing binding.

4.3 New IPv6 Destination Option

Mobile IPv6 defines a new IPv6 destination option, the Home Address destination option. This option is described in detail in Section 6.3.

4.4 New IPv6 ICMP Messages

Mobile IPv6 also introduces four new ICMP message types, two for use in the dynamic home agent address discovery mechanism, and two for renumbering and mobile configuration mechanisms. As described in Section 10.5 and Section 11.4.1, the following two new ICMP message types are used for home agent address discovery:

- o Home Agent Address Discovery Request, described in Section 6.5.
- o Home Agent Address Discovery Reply, described in Section 6.6.

The next two message types are used for network renumbering and address configuration on the mobile node, as described in Section 10.6:

- o Mobile Prefix Solicitation, described in Section 6.7.
- o Mobile Prefix Advertisement, described in Section 6.8.

4.5 Conceptual Data Structure Terminology

This document describes the Mobile IPv6 protocol in terms of the following conceptual data structures:

Binding Cache

A cache of bindings for other nodes. This cache is maintained by home agents and correspondent nodes. The cache contains both "correspondent registration" entries (see Section 9.1) and "home registration" entries (see Section 10.1).

Binding Update List

This list is maintained by each mobile node. The list has an item for every binding that the mobile node has or is trying to establish with a specific other node. Both correspondent and home

registrations are included in this list. Entries from the list are deleted as the lifetime of the binding expires. See Section 11.1.

Home Agents List

Home agents need to know which other home agents are on the same link. This information is stored in the Home Agents List, as described in more detail in Section 10.1. The list is used for informing mobile nodes during dynamic home agent address discovery.

4.6 Site-Local Addressability

This specification requires that home and care-of addresses **MUST** be unicast routable addresses. Site-local addresses may be usable on networks that are not connected to the Internet, but this specification does not define when such usage is safe and when not. Mobile nodes may not be aware of which site they are currently in, it is hard to prevent accidental attachment to other sites, and ambiguity of site-local addresses can cause problems if the home and visited networks use the same addresses. Therefore, site-local addresses **SHOULD NOT** be used as home or care-of addresses.

5. Overview of Mobile IPv6 Security

This specification provides a number of security features. These include the protection of Binding Updates both to home agents and correspondent nodes, the protection of mobile prefix discovery, and the protection of the mechanisms that Mobile IPv6 uses for transporting data packets.

Binding Updates are protected by the use of IPsec extension headers, or by the use of the Binding Authorization Data option. This option employs a binding management key, Kbm, which can be established through the return routability procedure. Mobile prefix discovery is protected through the use of IPsec extension headers. Mechanisms related to transporting payload packets – such as the Home Address destination option and type 2 routing header – have been specified in a manner which restricts their use in attacks.

5.1 Binding Updates to Home Agents

The mobile node and the home agent **MUST** use an IPsec security association to protect the integrity and authenticity of the Binding Updates and Acknowledgements. Both the mobile nodes and the home agents **MUST** support and **SHOULD** use the Encapsulating Security Payload (ESP) [6] header in transport mode and **MUST** use a non-NULL payload authentication algorithm to provide data origin authentication, connectionless integrity and optional anti-replay protection. Note that Authentication Header (AH) [5] is also possible but for brevity not discussed in this specification.

In order to protect messages exchanged between the mobile node and the home agent with IPsec, appropriate security policy database entries must be created. A mobile node must be prevented from using its security association to send a Binding Update on behalf of another mobile node using the same home agent. This **MUST** be achieved by having the home agent check that the given home address has been used with the right security association. Such a check is provided in the IPsec processing, by having the security policy database entries unequivocally identify a single security association for protecting Binding Updates between any given home address and home agent. In order to make this possible, it is necessary that the home address of the mobile node is visible in the Binding Updates and Acknowledgements. The home address is used in these packets as a source or destination, or in the Home Address Destination option or the type 2 routing header.

As with all IPsec security associations in this specification, manual configuration of security associations **MUST** be supported. The used shared secrets **MUST** be random and unique for different mobile nodes,

and MUST be distributed off-line to the mobile nodes.

Automatic key management with IKE [9] MAY be supported. When IKE is used, either the security policy database entries or the MIPv6 processing MUST unequivocally identify the IKE phase 1 credentials which can be used to authorize the creation of security associations for protecting Binding Updates for a particular home address. How these mappings are maintained is outside the scope of this specification, but they may be maintained, for instance, as a locally administered table in the home agent. If the phase 1 identity is a Fully Qualified Domain Name (FQDN), secure forms of DNS may also be used.

Section 11.3.2 discusses how IKE connections to the home agent need a careful treatment of the addresses used for transporting IKE. This is necessary to ensure that a Binding Update is not needed before the IKE exchange which is needed for securing the Binding Update.

When IKE version 1 is used with preshared secret authentication between the mobile node and the home agent, aggressive mode MUST be used. Similarly, the ID_IPV6_ADDR Identity Payload MUST NOT be used in IKEv1 phase 1.

Reference [21] contains a more detailed description and examples on using IPsec to protect the communications between the mobile node and the home agent.

5.2 Binding Updates to Correspondent Nodes

The protection of Binding Updates sent to correspondent nodes does not require the configuration of security associations or the existence of an authentication infrastructure between the mobile nodes and correspondent nodes. Instead, a method called the return routability procedure is used to assure that the right mobile node is sending the message. This method does not protect against attackers who are on the path between the home network and the correspondent node. However, attackers in such a location are capable of performing the same attacks even without Mobile IPv6. The main advantage of the return routability procedure is that it limits the potential attackers to those having an access to one specific path in the Internet, and avoids forged Binding Updates from anywhere else in the Internet. For a more in depth explanation of the security properties of the return routability procedure, see Section 15.

The integrity and authenticity of the Binding Updates messages to correspondent nodes is protected by using a keyed-hash algorithm. The binding management key, Kbm, is used to key the hash algorithm for this purpose. Kbm is established using data exchanged during the

return routability procedure. The data exchange is accomplished by use of node keys, nonces, cookies, tokens, and certain cryptographic functions. Section 5.2.5 outlines the basic return routability procedure. Section 5.2.6 shows how the results of this procedure are used to authorize a Binding Update to a correspondent node.

5.2.1 Node Keys

Each correspondent node has a secret key, K_{cn} , called the "node key", which it uses to produce the keygen tokens sent to the mobile nodes. The node key **MUST** be a random number, 20 octets in length. The node key allows the correspondent node to verify that the keygen tokens used by the mobile node in authorizing a Binding Update are indeed its own. This key **MUST NOT** be shared with any other entity.

A correspondent node **MAY** generate a fresh node key at any time; this avoids the need for secure persistent key storage. Procedures for optionally updating the node key are discussed later in Section 5.2.7.

5.2.2 Nonces

Each correspondent node also generates nonces at regular intervals. The nonces should be generated by using a random number generator that is known to have good randomness properties [1]. A correspondent node may use the same K_{cn} and nonce with all the mobiles it is in communication with.

Each nonce is identified by a nonce index. When a new nonce is generated, it must be associated with a new nonce index; this may be done, for example, by incrementing the value of the previous nonce index, if the nonce index is used as an array pointer into a linear array of nonces. However, there is no requirement that nonces be stored that way, or that the values of subsequent nonce indices have any particular relationship to each other. The index value is communicated in the protocol, so that if a nonce is replaced by new nonce during the run of a protocol, the correspondent node can distinguish messages that should be checked against the old nonce from messages that should be checked against the new nonce. Strictly speaking, indices are not necessary in the authentication, but allow the correspondent node to efficiently find the nonce value that it used in creating a keygen token.

Correspondent nodes keep both the current nonce and a small set of valid previous nonces whose lifetime has not yet expired. Expired values **MUST** be discarded, and messages using stale or unknown indices will be rejected.

The specific nonce index values cannot be used by mobile nodes to determine the validity of the nonce. Expected validity times for the nonces values and the procedures for updating them are discussed later in Section 5.2.7.

A nonce is an octet string of any length. The recommended length is 64 bits.

5.2.3 Cookies and Tokens

The return routability address test procedure uses cookies and keygen tokens as opaque values within the test init and test messages, respectively.

- o The "home init cookie" and "care-of init cookie" are 64 bit values sent to the correspondent node from the mobile node, and later returned to the mobile node. The home init cookie is sent in the Home Test Init message, and returned in the Home Test message. The care-of init cookie is sent in the Care-of Test Init message, and returned in the Care-of Test message.
- o The "home keygen token" and "care-of keygen token" are 64-bit values sent by the correspondent node to the mobile node via the home agent (via the Home Test message) and the care-of address (by the Care-of Test message), respectively.

The mobile node should set the home init or care-of init cookie to a newly generated random number in every Home or Care-of Test Init message it sends. The cookies are used to verify that the Home Test or Care-of Test message matches the Home Test Init or Care-of Test Init message, respectively. These cookies also serve to ensure that parties who have not seen the request cannot spoof responses.

Home and care-of keygen tokens are produced by the correspondent node based on its currently active secret key (Kcn) and nonces, as well as the home or care-of address (respectively). A keygen token is valid as long as both the secret key (Kcn) and the nonce used to create it are valid.

5.2.4 Cryptographic Functions

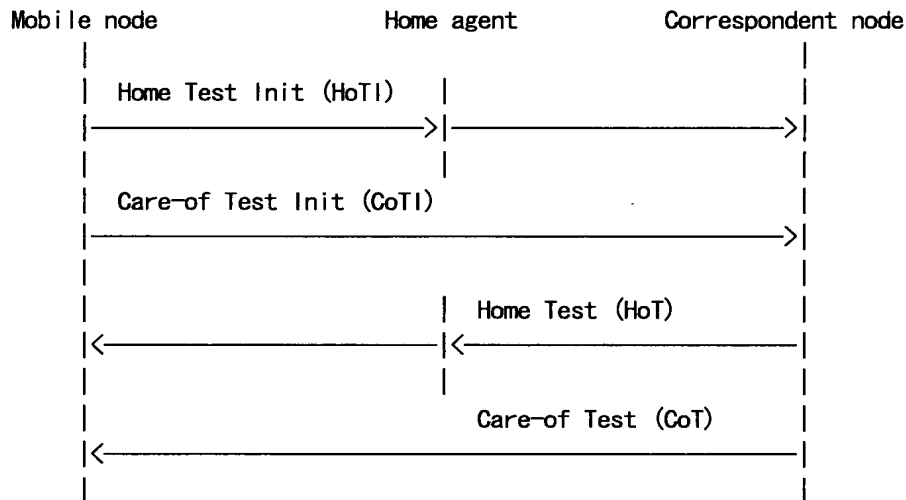
In this specification, the function used to compute hash values is SHA1 [20]. Message Authentication Codes (MACs) are computed using HMAC_SHA1 [25, 20]. $\text{HMAC_SHA1}(K, m)$ denotes such a MAC computed on message m with key K .

5.2.5 Return Routability Procedure

The Return Routability Procedure enables the correspondent node to obtain some reasonable assurance that the mobile node is in fact addressable at its claimed care-of address as well as at its home address. Only with this assurance is the correspondent node able to accept Binding Updates from the mobile node which would then instruct the correspondent node to direct that mobile node's data traffic to its claimed care-of address.

This is done by testing whether packets addressed to the two claimed addresses are routed to the mobile node. The mobile node can pass the test only if it is able to supply proof that it received certain data (the "keygen tokens") which the correspondent node sends to those addresses. These data are combined by the mobile node into a binding management key, denoted Kbm.

The below figure shows the message flow for the return routability procedure.



The Home and Care-of Test Init messages are sent at the same time. The procedure requires very little processing at the correspondent node, and the Home and Care-of Test messages can be returned quickly, perhaps nearly simultaneously. These four messages form the return routability procedure.

Home Test Init

A mobile node sends a Home Test Init message to the correspondent node (via the home agent) to acquire the home keygen token. The contents of the message can be summarized as follows:

- * Source Address = home address

- * Destination Address = correspondent
- * Parameters:
 - + home init cookie

The Home Test Init message conveys the mobile node's home address to the correspondent node. The mobile node also sends along a home init cookie that the correspondent node must return later. The Home Test Init message is reverse tunneled through the home agent. (The headers and addresses related to reverse tunneling have been omitted from the above discussion of the message contents.) The mobile node remembers these cookie values to obtain some assurance that its protocol messages are being processed by the desired correspondent node.

Care-of Test Init

The mobile node sends a Care-of Test Init message to the correspondent node (directly, not via the home agent) to acquire the care-of keygen token. The contents of this message can be summarized as follows:

- * Source Address = care-of address
- * Destination Address = correspondent
- * Parameters:
 - + care-of init cookie

The Care-of Test Init message conveys the mobile node's care-of address to the correspondent node. The mobile node also sends along a care-of init cookie that the correspondent node must return later. The Care-of Test Init message is sent directly to the correspondent node.

Home Test

The Home Test message is sent in response to a Home Test Init message. It is sent via the home agent. The contents of the message are:

- * Source Address = correspondent

- * Destination Address = home address

- * Parameters:

- + home init cookie
- + home keygen token
- + home nonce index

When the correspondent node receives the Home Test Init message, it generates a home keygen token as follows:

home keygen token :=
First (64, HMAC_SHA1 (Kcn, (home address | nonce | 0)))

where | denotes concatenation. The final "0" inside the HMAC_SHA1 function is a single zero octet, used to distinguish home and care-of cookies from each other.

The home keygen token is formed from the first 64 bits of the MAC. The home keygen token tests that the mobile node can receive messages sent to its home address. Kcn is used in the production of home keygen token in order to allow the correspondent node to verify that it generated the home and care-of nonces, without forcing the correspondent node to remember a list of all tokens it has handed out.

The Home Test message is sent to the mobile node via the home network, where it is presumed that the home agent will tunnel the message to the mobile node. This means that the mobile node needs to already have sent a Binding Update to the home agent, so that the home agent will have received and authorized the new care-of address for the mobile node before the return routability procedure. For improved security, the data passed between the home agent and the mobile node is made immune to inspection and passive attacks. Such protection is gained by encrypting the home keygen token as it is tunneled from the home agent to the mobile node as specified in Section 10.4.6. The security properties of this additional security are discussed in Section 15.4.1.

The home init cookie from the mobile node is returned in the Home Test message, to ensure that the message comes from a node on the route between the home agent and the correspondent node.

The home nonce index is delivered to the mobile node to later

allow the correspondent node to efficiently find the nonce value that it used in creating the home keygen token.

Care-of Test

This message is sent in response to a Care-of Test Init message. This message is not sent via the home agent, it is sent directly to the mobile node. The contents of the message are:

- * Source Address = correspondent
- * Destination Address = care-of address
- * Parameters:
 - + care-of init cookie
 - + care-of keygen token
 - + care-of nonce index

When the correspondent node receives the Care-of Test Init message, it generates a care-of keygen token as follows:

care-of keygen token :=
First (64, HMAC_SHA1 (Kcn, (care-of address | nonce | 1)))

Here, the final "1" inside the HMAC_SHA1 function is a single octet containing the hex value 0x01, and is used to distinguish home and care-of cookies from each other. The keygen token is formed from the first 64 bits of the MAC, and sent directly to the mobile node at its care-of address. The care-of init cookie from the Care-of Test Init message is returned to ensure that the message comes from a node on the route to the correspondent node.

The care-of nonce index is provided to identify the nonce used for the care-of keygen token. The home and care-of nonce indices MAY be the same, or different, in the Home and Care-of Test messages.

When the mobile node has received both the Home and Care-of Test messages, the return routability procedure is complete. As a result of the procedure, the mobile node has the data it needs to send a Binding Update to the correspondent node. The mobile node hashes the tokens together to form a 20 octet binding key Kbm:

$K_{bm} = \text{SHA1}(\text{home keygen token} \parallel \text{care-of keygen token})$

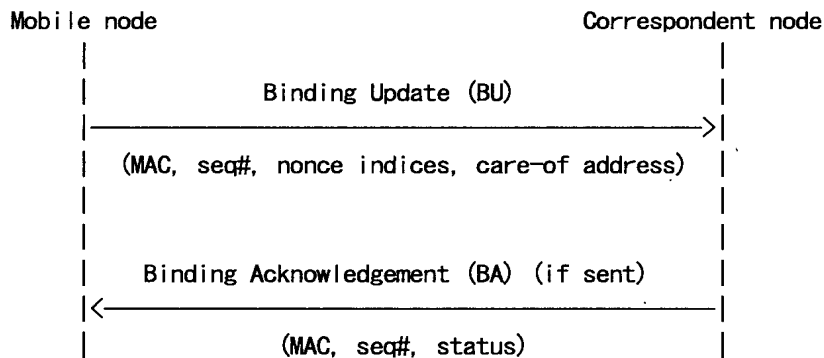
A Binding Update may also be used to delete a previously established binding (Section 6.1.7). In this case, the care-of keygen token is not used. Instead, the binding management key is generated as follows:

$K_{bm} = \text{SHA1}(\text{home keygen token})$

Note that the correspondent node does not create any state specific to the mobile node, until it receives the Binding Update from that mobile node. The correspondent node does not maintain the value for the binding management key K_{bm} ; it creates K_{bm} when given the nonce indices and the mobile node's addresses.

5.2.6 Authorizing Binding Management Messages

After the mobile node has created the binding management key (K_{bm}), it can supply a verifiable Binding Update to the correspondent node. This section provides an overview of this registration. The below figure shows the message flow.



Binding Update

To authorize a Binding Update, the mobile node creates a binding management key K_{bm} from the keygen tokens as described in the previous section. The contents of the Binding Update include the following:

- * Source Address = care-of address
- * Destination Address = correspondent
- * Parameters:
 - + home address (within the Home Address destination option if

different from the Source Address)

- + sequence number (within the Binding Update message header)
- + home nonce index (within the Nonce Indices option)
- + care-of nonce index (within the Nonce Indices option)
- + First (96, HMAC_SHA1 (Kbm, (care-of address | correspondent | BU)))

The Binding Update contains a Nonce Indices option, indicating to the correspondent node which home and care-of nonces to use to recompute Kbm, the binding management key. The MAC is computed as described in Section 6.2.7, using the correspondent node's address as the destination address and the Binding Update message itself ("BU" above) as the MH Data.

Once the correspondent node has verified the MAC, it can create a Binding Cache entry for the mobile.

Binding Acknowledgement

The Binding Update is in some cases acknowledged by the correspondent node. The contents of the message are as follows:

- * Source Address = correspondent
- * Destination Address = care-of address
- * Parameters:
 - + sequence number (within the Binding Update message header)
 - + First (96, HMAC_SHA1 (Kbm, (care-of address | correspondent | BA)))

The Binding Acknowledgement contains the same sequence number as the Binding Update. The MAC is computed as described in Section 6.2.7, using the correspondent node's address as the destination address and the message itself ("BA" above) as the MH Data.

Bindings established with correspondent nodes using keys created by way of the return routability procedure MUST NOT exceed MAX_RR_BINDING_LIFETIME seconds (see Section 12).

The value in the Source Address field in the IPv6 header carrying the Binding Update is normally also the care-of address which is used in the binding. However, a different care-of address **MAY** be specified by including an Alternate Care-of Address mobility option in the Binding Update (see Section 6.2.5). When such a message is sent to the correspondent node and the return routability procedure is used as the authorization method, the Care-of Test Init and Care-of Test messages **MUST** have been performed for the address in the Alternate Care-of Address option (not the Source Address). The nonce indices and MAC value **MUST** be based on information gained in this test.

Binding Updates may also be sent to delete a previously established binding. In this case, generation of the binding management key depends exclusively on the home keygen token and the care-of nonce index is ignored.

5.2.7 Updating Node Keys and Nonces

Correspondent nodes generate nonces at regular intervals. It is recommended to keep each nonce (identified by a nonce index) acceptable for at least `MAX_TOKEN_LIFETIME` seconds (see Section 12) after it has been first used in constructing a return routability message response. However, the correspondent node **MUST NOT** accept nonces beyond `MAX_NONCE_LIFETIME` seconds (see Section 12) after the first use. As the difference between these two constants is 30 seconds, a convenient way to enforce the above lifetimes is to generate a new nonce every 30 seconds. The node can then continue to accept tokens that have been based on the last 8 ($\text{MAX_NONCE_LIFETIME} / 30$) nonces. This results in tokens being acceptable `MAX_TOKEN_LIFETIME` to `MAX_NONCE_LIFETIME` seconds after they have been sent to the mobile node, depending on whether the token was sent at the beginning or end of the first 30 second period. Note that the correspondent node may also attempt to generate new nonces on demand, or only if the old nonces have been used. This is possible, as long as the correspondent node keeps track of how long a time ago the nonces were used for the first time, and does not generate new nonces on every return routability request.

Due to resource limitations, rapid deletion of bindings, or reboots the correspondent node may not in all cases recognize the nonces that the tokens were based on. If a nonce index is unrecognized, the correspondent node replies with an error code in the Binding Acknowledgement (either 136, 137, or 138 as discussed in Section 6.1.8). The mobile node can then retry the return routability procedure.

An update of Kcn **SHOULD** be done at the same time as an update of a nonce, so that nonce indices can identify both the nonce and the key.

Old Kcn values have to be therefore remembered as long as old nonce values.

Given that the tokens are normally expected to be usable for MAX_TOKEN_LIFETIME seconds, the mobile node MAY use them beyond a single run of the return routability procedure until MAX_TOKEN_LIFETIME expires. After this the mobile node SHOULD NOT use the tokens. A fast moving mobile node MAY reuse a recent home keygen token from a correspondent node when moving to a new location, and just acquire a new care-of keygen token to show routability in the new location.

While this does not save the number of round-trips due to the simultaneous processing of home and care-of return routability tests, there are fewer messages being exchanged, and a potentially long round-trip through the home agent is avoided. Consequently, this optimization is often useful. A mobile node that has multiple home addresses, MAY also use the same care-of keygen token for Binding Updates concerning all of these addresses.

5.2.8 Preventing Replay Attacks

The return routability procedure also protects the participants against replayed Binding Updates through the use of the sequence number and a MAC. Care must be taken when removing bindings at the correspondent node, however. Correspondent nodes must retain bindings and the associated sequence number information at least as long as the nonces used in the authorization of the binding are still valid. Alternatively, if memory is very constrained, the correspondent node MAY invalidate the nonces that were used for the binding being deleted (or some larger group of nonces that they belong to). This may, however, impact the ability to accept Binding Updates from mobile nodes that have recently received keygen tokens. This alternative is therefore recommended only as a last measure.

5.3 Dynamic Home Agent Address Discovery

No security is required for dynamic home agent address discovery.

5.4 Mobile Prefix Discovery

The mobile node and the home agent SHOULD use an IPsec security association to protect the integrity and authenticity of the Mobile Prefix Solicitations and Advertisements. Both the mobile nodes and the home agents MUST support and SHOULD use the Encapsulating Security Payload (ESP) header in transport mode with a non-NULL payload authentication algorithm to provide data origin authentication, connectionless integrity and optional anti-replay

protection.

5.5 Payload Packets

Payload packets exchanged with mobile nodes can be protected in the usual manner, in the same way as stationary hosts can protect them. However, Mobile IPv6 introduces the Home Address destination option, a routing header, and tunneling headers in the payload packets. In the following we define the security measures taken to protect these, and to prevent their use in attacks against other parties.

This specification limits the use of the Home Address destination option to the situation where the correspondent node already has a Binding Cache entry for the given home address. This avoids the use of the Home Address option in attacks described in Section 15.1.

Mobile IPv6 uses a Mobile IPv6 specific type of a routing header. This type provides the necessary functionality but does not open vulnerabilities discussed in Section 15.1.

Tunnels between the mobile node and the home agent are protected by ensuring proper use of source addresses, and optional cryptographic protection. The mobile node verifies that the outer IP address corresponds to its home agent. The home agent verifies that the outer IP address corresponds to the current location of the mobile node (Binding Updates sent to the home agents are secure). The home agent identifies the mobile node through the source address of the inner packet. (Typically, this is the home address of the mobile node, but it can also be a link-local address, as discussed in Section 10.4.2. To recognize the latter type of addresses, the home agent requires that the Link-Local Address Compatibility (L) was set in the Binding Update.) These measures protect the tunnels against vulnerabilities discussed in Section 15.1.

For traffic tunneled via the home agent, additional IPsec ESP encapsulation MAY be supported and used. If multicast group membership control protocols or stateful address autoconfiguration protocols are supported, payload data protection MUST be supported.

6. New IPv6 Protocol, Message Types, and Destination Option

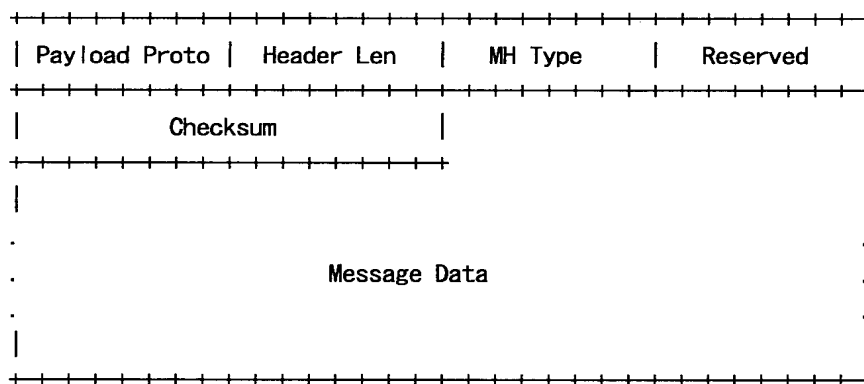
6.1 Mobility Header

The Mobility Header is an extension header used by mobile nodes, correspondent nodes, and home agents in all messaging related to the creation and management of bindings. The subsections within this section describe the message types that may be sent using the Mobility Header.

Mobility Header messages **MUST NOT** be sent with a type 2 routing header, except as described in Section 9.5.4 for Binding Acknowledgement. Mobility Header messages also **MUST NOT** be used with a Home Address destination option, except as described in Section 11.7.1 and Section 11.7.2 for Binding Update. Binding Update List or Binding Cache information (when present) for the destination **MUST NOT** be used in sending Mobility Header messages. That is, Mobility Header messages bypass both the Binding Cache check described in Section 9.3.2 and the Binding Update List check described in Section 11.3.1 which are normally performed for all packets. This applies even to messages sent to or from a correspondent node which is itself a mobile node.

6.1.1 Format

The Mobility Header is identified by a Next Header value of TBD <To be assigned by IANA> in the immediately preceding header, and has the following format:



Payload Proto

8-bit selector. Identifies the type of header immediately following the Mobility Header. Uses the same values as the IPv6 Next Header field [11].

This field is intended to be used by a future extension (see

Appendix B.1).

Implementations conforming to this specification SHOULD set the payload protocol type to IPPROTO_NONE (59 decimal).

Header Len

8-bit unsigned integer, representing the length of the Mobility Header in units of 8 octets, excluding the first 8 octets.

The length of the Mobility Header MUST be a multiple of 8 octets.

MH Type

8-bit selector. Identifies the particular mobility message in question. Current values are specified in Section 6.1.2 and onward. An unrecognized MH Type field causes an error indication to be sent.

Reserved

8-bit field reserved for future use. The value MUST be initialized to zero by the sender, and MUST be ignored by the receiver.

Checksum

16-bit unsigned integer. This field contains the checksum of the Mobility Header. The checksum is calculated from the octet string consisting of a "pseudo-header" followed by the entire Mobility Header starting with the Payload Proto field. The checksum is the 16-bit one's complement of the one's complement sum of this string.

The pseudo-header contains IPv6 header fields, as specified in Section 8.1 of RFC 2460 [11]. The Next Header value used in the pseudo-header is TBD <To be assigned by IANA>. The addresses used in the pseudo-header are the addresses that appear in the Source and Destination Address fields in the IPv6 packet carrying the Mobility Header.

Note that the procedures of calculating upper layer checksums while away from home described in Section 11.3.1 apply even for the Mobility Header. If a mobility message has a Home Address destination option, then the checksum calculation uses the home address in this option as the value of the IPv6 Source Address field. The type 2 routing header is treated as explained in [11].

For computing the checksum, the checksum field is set to zero.

A variable length field containing the data specific to the indicated Mobility Header type.

Alignment requirements for the Mobility Header are the same as for any IPv6 protocol Header. That is, they **MUST** be aligned on an 8-octet boundary.

The Binding Refresh Request (BRR) message requests a mobile node to update its mobility binding. This message is sent by correspondent nodes according to the rules in Section 9.5.5. When a mobile node receives a packet containing a Binding Refresh Request message it processes the message according to the rules in Section 11.7.4.

The diagram shows a horizontal line representing a 64-bit address. The top 16 bits are marked with a vertical line and the word "Reserved". The bottom 48 bits are marked with a vertical line and the words "Mobility options".

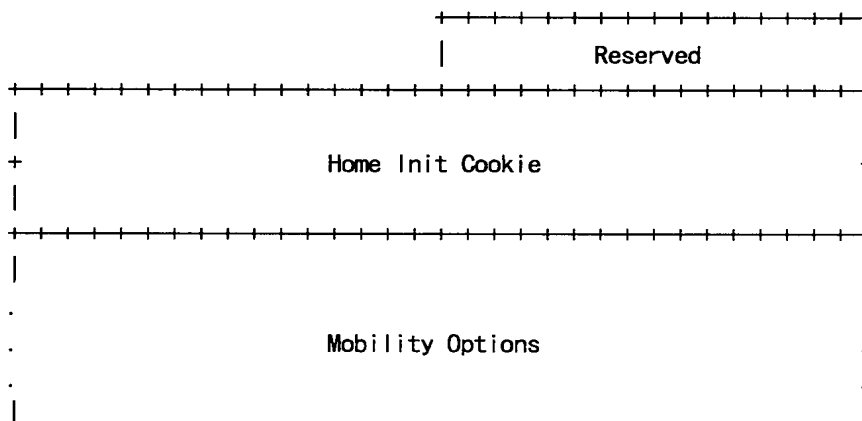
16-bit field reserved for future use. The value **MUST** be initialized to zero by the sender, and **MUST** be ignored by the receiver.

Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The encoding and format of defined options are described in Section 6.2. The receiver **MUST** ignore and skip any options which it does not understand.

There MAY be additional information, associated with this Binding Refresh Request message that need not be present in all Binding Refresh Request messages sent. Mobility options allow future extensions to the format of the Binding Refresh Request message to be defined. This specification does not define any options valid for the Binding Refresh Request message.

6.1.3 Home Test Init Message

A mobile node uses the Home Test Init (HoTI) message to initiate the return routability procedure and request a home keygen token from a correspondent node (see Section 11.6.1). The Home Test Init message uses the MH Type value 1. When this value is indicated in the MH Type field, the format of the Message Data field in the Mobility Header is as follows:



Reserved

16-bit field reserved for future use. This value **MUST** be initialized to zero by the sender, and **MUST** be ignored by the receiver.

Home Init Cookie

64-bit field which contains a random value, the home init cookie.

Mobility Options

Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The receiver **MUST** ignore and skip any options which it does not understand. This specification does not define any options valid for the Home Test Init message.

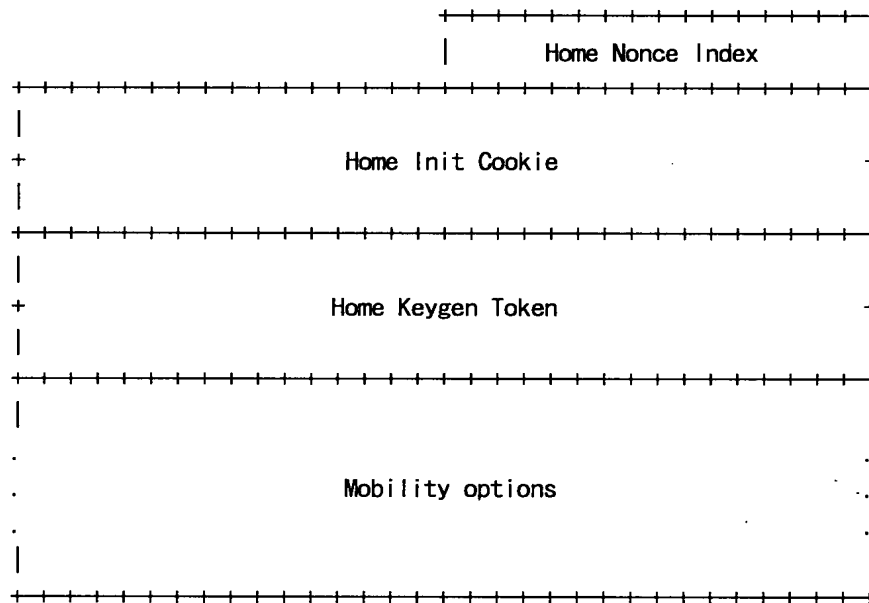
If no actual options are present in this message, no padding is necessary and the Header Len field will be set to 1.

This message is tunneled through the home agent when the mobile node is away from home. Such tunneling **SHOULD** employ IPsec ESP in tunnel mode between the home agent and the mobile node. This protection is indicated by the IPsec security policy database. The protection of Home Test Init messages is unrelated to the requirement to protect regular payload traffic, which **MAY** use such tunnels as well.

6.1.4 Care-of Test Init Message

A mobile node uses the Care-of Test Init (CoTI) message to initiate the return routability procedure and request a care-of keygen token from a correspondent node (see Section 11.6.1). The Care-of Test Init message uses the MH Type value 2. When this value is indicated in the MH Type field, the format of the Message Data field in the Mobility Header is as follows:





Home Nonce Index

This field will be echoed back by the mobile node to the correspondent node in a subsequent Binding Update.

Home Init Cookie

64-bit field which contains the home init cookie.

Home Keygen Token

This field contains the 64 bit home keygen token used in the return routability procedure.

Mobility Options

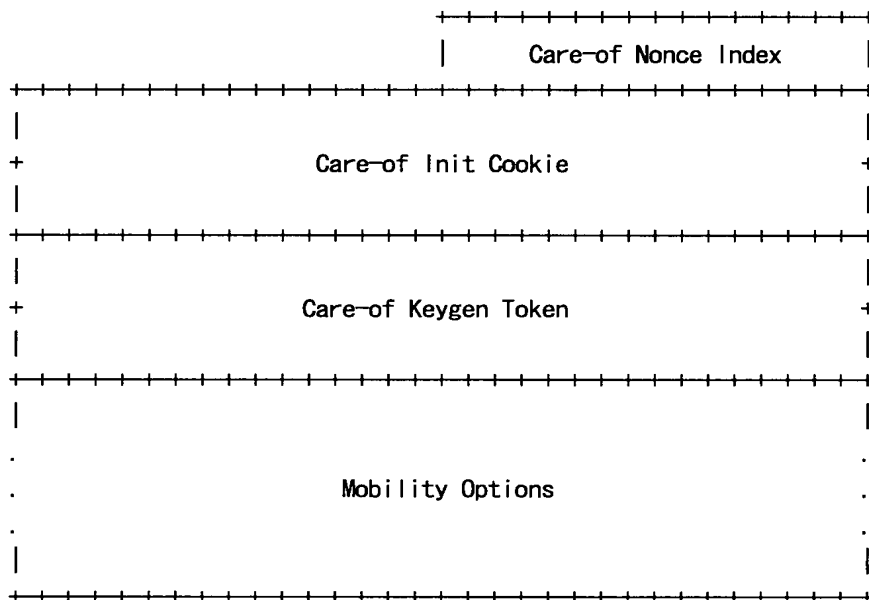
Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The receiver MUST ignore and skip any options which it does not understand. This specification does not define any options valid for the Home Test message.

If no actual options are present in this message, no padding is necessary and the Header Len field will be set to 2.

6.1.6 Care-of Test Message

The Care-of Test (CoT) message is a response to the Care-of Test Init message, and is sent from the correspondent node to the mobile node

(see Section 11.6.2). The Care-of Test message uses the MH Type value 4. When this value is indicated in the MH Type field, the format of the Message Data field in the Mobility Header is as follows:



Care-of Nonce Index

This value will be echoed back by the mobile node to the correspondent node in a subsequent Binding Update.

Care-of Init Cookie

64-bit field which contains the care-of init cookie.

Care-of Keygen Token

This field contains the 64 bit care-of keygen token used in the return routability procedure.

Mobility Options

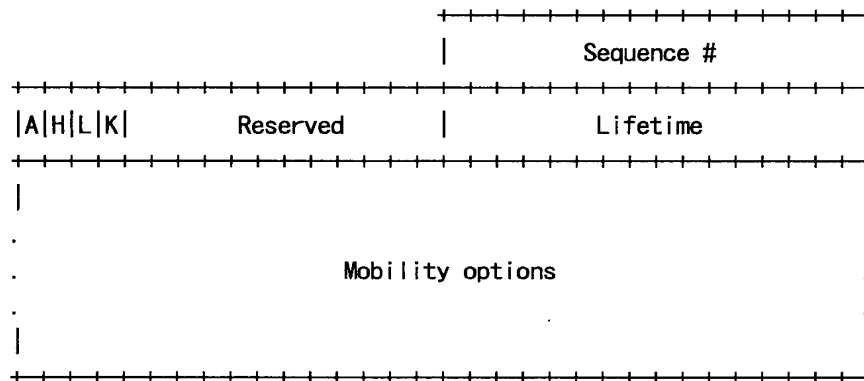
Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The receiver MUST ignore and skip any options which it does not understand. This specification does not define any options valid for the Care-of Test message.

If no actual options are present in this message, no padding is necessary and the Header Len field will be set to 2.

6.1.7 Binding Update Message

The Binding Update (BU) message is used by a mobile node to notify other nodes of a new care-of address for itself. Binding Updates are sent as described in Section 11.7.1 and Section 11.7.2.

The Binding Update uses the MH Type value 5. When this value is indicated in the MH Type field, the format of the Message Data field in the Mobility Header is as follows:



Acknowledge (A)

The Acknowledge (A) bit is set by the sending mobile node to request a Binding Acknowledgement (Section 6.1.8) be returned upon receipt of the Binding Update.

Home Registration (H)

The Home Registration (H) bit is set by the sending mobile node to request that the receiving node should act as this node's home agent. The destination of the packet carrying this message **MUST** be that of a router sharing the same subnet prefix as the home address of the mobile node in the binding.

Link-Local Address Compatibility (L)

The Link-Local Address Compatibility (L) bit is set when the home address reported by the mobile node has the same interface identifier as the mobile node's link-local address.

Key Management Mobility Capability (K)

If this bit is cleared, the protocol used for establishing the IPsec security associations between the mobile node and the home agent does not survive movements. It may then have to be rerun. (Note that the IPsec security associations themselves are expected

to survive movements.) If manual IPsec configuration is used, the bit **MUST** be cleared.

This bit is valid only in Binding Updates sent to the home agent, and **MUST** be cleared in other Binding Updates. Correspondent nodes **MUST** ignore this bit.

Reserved

These fields are unused. They **MUST** be initialized to zero by the sender and **MUST** be ignored by the receiver.

Sequence

A 16-bit unsigned integer used by the receiving node to sequence Binding Updates and by the sending node to match a returned Binding Acknowledgement with this Binding Update.

Lifetime

16-bit unsigned integer. The number of time units remaining before the binding **MUST** be considered expired. A value of zero indicates that the Binding Cache entry for the mobile node **MUST** be deleted. (In this case the specified care-of address **MUST** also be set equal to the home address.) One time unit is 4 seconds.

Mobility Options

Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The encoding and format of defined options are described in Section 6.2. The receiver **MUST** ignore and skip any options which it does not understand.

The following options are valid in a Binding Update:

- * Binding Authorization Data option (this option is mandatory in Binding Updates sent to a correspondent node)
- * Nonce Indices option.
- * Alternate Care-of Address option

If no options are present in this message, 4 octets of padding is necessary and the Header Len field will be set to 1.

The care-of address is specified either by the Source Address field

in the IPv6 header or by the Alternate Care-of Address option, if present. The care-of address **MUST** be a unicast routable address. IPv6 Source Address **MUST** be a topologically correct source address. Binding Updates for a care-of address which is not a unicast routable address **MUST** be silently discarded. Similarly, the Binding Update **MUST** be silently discarded if the care-of address appears as a home address in an existing Binding Cache entry, with its current location creating a circular reference back to the home address specified in the Binding Update (possibly through additional entries).

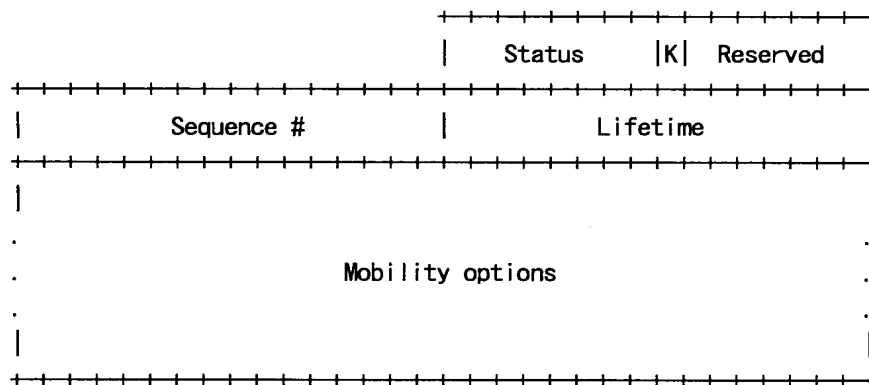
The deletion of a binding can be indicated by setting the Lifetime field to 0 and by setting the care-of address equal to the home address. In deletion, the generation of the binding management key depends exclusively on the home keygen token, as explained in Section 5.2.5. (Note that while the senders are required to set both the Lifetime field to 0 and the care-of address equal to the home address, Section 9.5.1 rules for receivers are more liberal, and interpret either condition as a deletion.)

Correspondent nodes **SHOULD NOT** expire the Binding Cache entry before the lifetime expires, if any application hosted by the correspondent node is still likely to require communication with the mobile node. A Binding Cache entry that is deallocated prematurely might cause subsequent packets to be dropped from the mobile node, if they contain the Home Address destination option. This situation is recoverable, since a Binding Error message is sent to the mobile node (see Section 6.1.9); however, it causes unnecessary delay in the communications.

6.1.8 Binding Acknowledgement Message

The Binding Acknowledgement is used to acknowledge receipt of a Binding Update (Section 6.1.7). This packet is sent as described in Section 9.5.4 and Section 10.3.1.

The Binding Acknowledgement has the MH Type value 6. When this value is indicated in the MH Type field, the format of the Message Data field in the Mobility Header is as follows:



Key Management Mobility Capability (K)

If this bit is cleared, the protocol used by the home agent for establishing the IPsec security associations between the mobile node and the home agent does not survive movements. It may then have to be rerun. (Note that the IPsec security associations themselves are expected to survive movements.)

Correspondent nodes **MUST** set the K bit to 0.

Reserved

These fields are unused. They **MUST** be initialized to zero by the sender and **MUST** be ignored by the receiver.

Status

8-bit unsigned integer indicating the disposition of the Binding Update. Values of the Status field less than 128 indicate that the Binding Update was accepted by the receiving node. Values greater than or equal to 128 indicate that the Binding Update was rejected by the receiving node. The following Status values are currently defined:

- 0 Binding Update accepted
- 1 Accepted but prefix discovery necessary
- 128 Reason unspecified
- 129 Administratively prohibited
- 130 Insufficient resources

- 131 Home registration not supported
- 132 Not home subnet
- 133 Not home agent for this mobile node
- 134 Duplicate Address Detection failed
- 135 Sequence number out of window
- 136 Expired home nonce index
- 137 Expired care-of nonce index
- 138 Expired nonces
- 139 Registration type change disallowed

Up-to-date values of the Status field are to be specified in the IANA registry of assigned numbers [19].

Sequence

The Sequence Number in the Binding Acknowledgement is copied from the Sequence Number field in the Binding Update. It is used by the mobile node in matching this Binding Acknowledgement with an outstanding Binding Update.

Lifetime

The granted lifetime, in time units of 4 seconds, for which this node **SHOULD** retain the entry for this mobile node in its Binding Cache.

The value of this field is undefined if the Status field indicates that the Binding Update was rejected.

Mobility Options

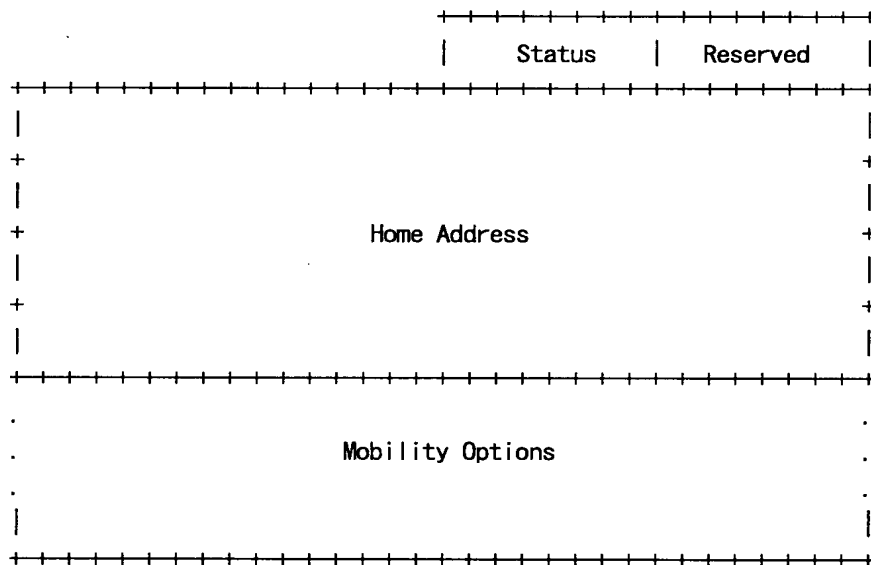
Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The encoding and format of defined options are described in Section 6.2. The receiver **MUST** ignore and skip any options which it does not understand.

There **MAY** be additional information, associated with this Binding

- * Binding Authorization Data option (this option is mandatory in Binding Acknowledgements sent by a correspondent node, except where otherwise noted in Section 9.5.4)
- * Binding Refresh Advice option

6.1.9 Binding Error Message

The Binding Error message uses the MH Type value 7. When this value is indicated in the MH Type field, the format of the Message Data field in the Mobility Header is as follows:



8-bit unsigned integer indicating the reason for this message.
The following values are currently defined:

1 Unknown binding for Home Address destination option

2 Unrecognized MH Type value

Reserved

A 8-bit field reserved for future use. The value **MUST** be initialized to zero by the sender, and **MUST** be ignored by the receiver.

Home Address

The home address that was contained in the Home Address destination option. The mobile node uses this information to determine which binding does not exist, in cases where the mobile node has several home addresses.

Mobility Options

Variable-length field of such length that the complete Mobility Header is an integer multiple of 8 octets long. This field contains zero or more TLV-encoded mobility options. The receiver **MUST** ignore and skip any options which it does not understand.

There **MAY** be additional information, associated with this Binding Error message that need not be present in all Binding Error messages sent. Mobility options allow future extensions to the format of the Binding Error message to be defined. The encoding and format of defined options are described in Section 6.2. This specification does not define any options valid for the Binding Error message.

If no actual options are present in this message, no padding is necessary and the Header Len field will be set to 2.

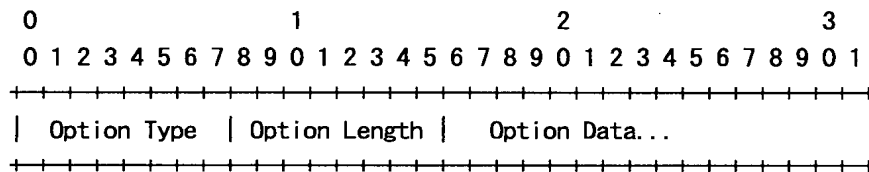
6.2 Mobility Options

Mobility messages can include zero or more mobility options. This allows optional fields that may not be needed in every use of a particular Mobility Header, as well as future extensions to the format of the messages. Such options are included in the Message Data field of the message itself, after the fixed portion of the message data specified in the message subsections of Section 6.1.

The presence of such options will be indicated by the Header Len of the Mobility Header. If included, the Binding Authorization Data option (Section 6.2.7) **MUST** be the last option and **MUST NOT** have trailing padding. Otherwise, options can be placed in any order.

6.2.1 Format

Mobility options are encoded within the remaining space of the Message Data field of a mobility message, using a type-length-value (TLV) format as follows:



Option Type

8-bit identifier of the type of mobility option. When processing a Mobility Header containing an option for which the Option Type value is not recognized by the receiver, the receiver **MUST** quietly ignore and skip over the option, correctly handling any remaining options in the message.

Option Length

8-bit unsigned integer, representing the length in octets of the mobility option, not including the Option Type and Option Length fields.

Option Data

A variable length field that contains data specific to the option.

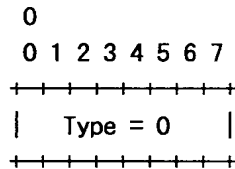
The following subsections specify the Option types which are currently defined for use in the Mobility Header.

Implementations **MUST** silently ignore any mobility options that they do not understand.

Mobility options may have alignment requirements. Following the convention in IPv6, these options are aligned in a packet so that multi-octet values within the Option Data field of each option fall on natural boundaries (i.e., fields of width *n* octets are placed at an integer multiple of *n* octets from the start of the header, for *n* = 1, 2, 4, or 8) [11].

6.2.2 Pad1

The Pad1 option does not have any alignment requirements. Its format is as follows:

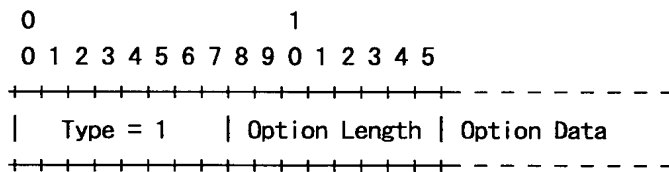


NOTE! the format of the Pad1 option is a special case – it has neither Option Length nor Option Data fields.

The Pad1 option is used to insert one octet of padding in the Mobility Options area of a Mobility Header. If more than one octet of padding is required, the PadN option, described next, should be used rather than multiple Pad1 options.

6.2.3 PadN

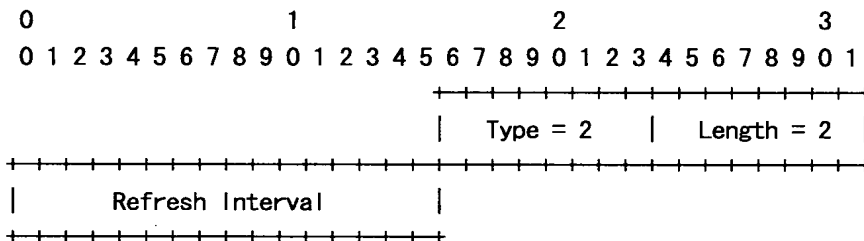
The PadN option does not have any alignment requirements. Its format is as follows:



The PadN option is used to insert two or more octets of padding in the Mobility Options area of a mobility message. For N octets of padding, the Option Length field contains the value N-2, and the Option Data consists of N-2 zero-valued octets. PadN Option data MUST be ignored by the receiver.

6.2.4 Binding Refresh Advice

The Binding Refresh Advice option has an alignment requirement of 2n. Its format is as follows:

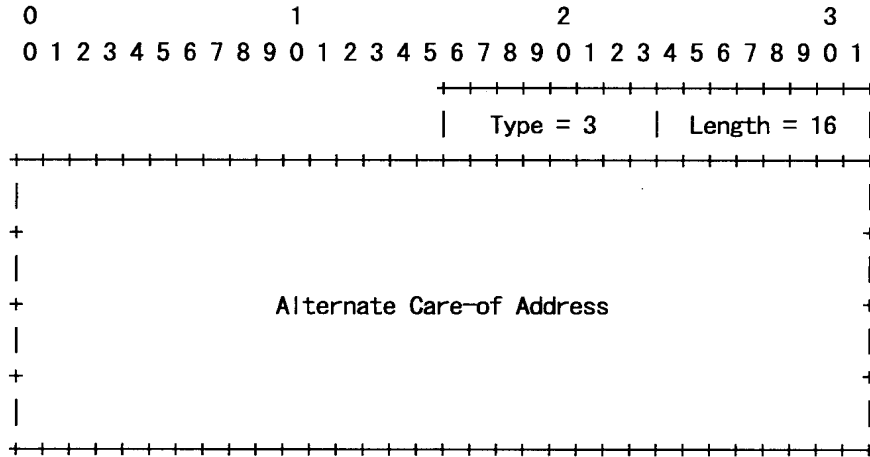


The Binding Refresh Advice option is only valid in the Binding Acknowledgement, and only on Binding Acknowledgements sent from the mobile node's home agent in reply to a home registration. The Refresh Interval is measured in units of four seconds, and indicates

how long before the mobile node SHOULD send a new home registration to the home agent. The Refresh Interval MUST be set to indicate a smaller time interval than the Lifetime value of the Binding Acknowledgement.

6.2.5 Alternate Care-of Address

The Alternate Care-of Address option has an alignment requirement of $8n+6$. Its format is as follows:

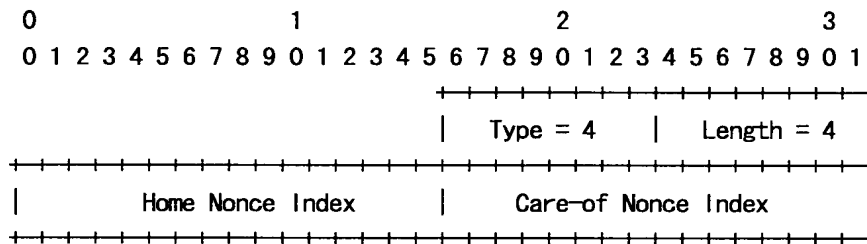


Normally, a Binding Update specifies the desired care-of address in the Source Address field of the IPv6 header. However, this is not possible in some cases, such as when the mobile node wishes to indicate a care-of address which it cannot use as a topologically correct source address (Section 6.1.7 and Section 11.7.2) or when the used security mechanism does not protect the IPv6 header (Section 11.7.1).

The Alternate Care-of Address option is provided for these situations. This option is valid only in Binding Update. The Alternate Care-of Address field contains an address to use as the care-of address for the binding, rather than using the Source Address of the packet as the care-of address.

6.2.6 Nonce Indices

The Nonce Indices option has an alignment requirement of $2n$. Its format is as follows:



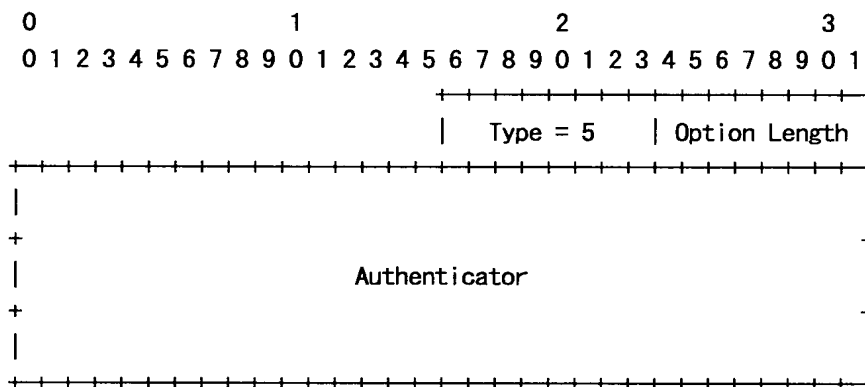
The Nonce Indices option is valid only in the Binding Update message sent to a correspondent node, and only when present together with a Binding Authorization Data option. When the correspondent node authorizes the Binding Update, it needs to produce home and care-of keygen tokens from its stored random nonce values.

The Home Nonce Index field tells the correspondent node which nonce value to use when producing the home keygen token.

The Care-of Nonce Index field is ignored in requests to delete a binding. Otherwise, it tells the correspondent node which nonce value to use when producing the care-of keygen token.

6.2.7 Binding Authorization Data

The Binding Authorization Data option does not have alignment requirements as such. However, since this option must be the last mobility option, an implicit alignment requirement is $8n + 2$. The format of this option is as follows:



The Binding Authorization Data option is valid in the Binding Update and Binding Acknowledgement.

The Option Length field contains the length of the authenticator in octets.

The Authenticator field contains a cryptographic value which can be used to determine that the message in question comes from the right

authority. Rules for calculating this value depend on the used authorization procedure.

For the return routability procedure, this option can appear in the Binding Update and Binding Acknowledgements. Rules for calculating the Authenticator value are the following:

$$\begin{aligned} \text{Mobility Data} &= \text{care-of address} \mid \text{correspondent} \mid \text{MH Data} \\ \text{Authenticator} &= \text{First (96, HMAC_SHA1 (Kbm, Mobility Data))} \end{aligned}$$

Where \mid denotes concatenation and "correspondent" is the IPv6 address of the correspondent node. Note that, if the message is sent to a destination which is itself mobile, the "correspondent" address may not be the address found in the Destination Address field of the IPv6 header; instead the home address from the type 2 Routing header should be used.

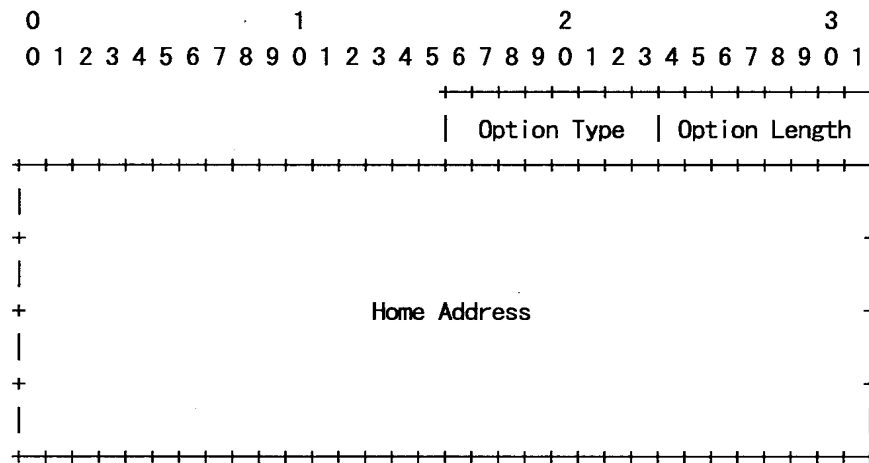
"MH Data" is the content of the Mobility Header, excluding the Authenticator field itself. The Authenticator value is calculated as if the Checksum field in the Mobility Header was zero. The Checksum in the transmitted packet is still calculated in the usual manner, with the calculated Authenticator being a part of the packet protected by the Checksum. Kbm is the binding management key, which is typically created using nonces provided by the correspondent node (see Section 9.4). Note that while the contents of a potential Home Address destination option are not covered in this formula, the rules for the calculation of the Kbm do take the home address in account. This ensures that the MAC will be different for different home addresses.

The first 96 bits from the MAC result are used as the Authenticator field.

6.3 Home Address Option

The Home Address option is carried by the Destination Option extension header (Next Header value = 60). It is used in a packet sent by a mobile node while away from home, to inform the recipient of the mobile node's home address.

The Home Address option is encoded in type-length-value (TLV) format as follows:



Option Type

201 = 0xC9

Option Length

8-bit unsigned integer. Length of the option, in octets, excluding the Option Type and Option Length fields. This field **MUST** be set to 16.

Home Address

The home address of the mobile node sending the packet. This address **MUST** be a unicast routable address.

The alignment requirement [11] for the Home Address option is $8n+6$.

The three highest-order bits of the Option Type field are encoded to indicate specific processing of the option [11]; for the Home Address option, these three bits are set to 110. This indicates the following processing requirements:

- o Any IPv6 node that does not recognize the Option Type must discard the packet, and if the packet's Destination Address was not a multicast address, return an ICMP Parameter Problem, Code 2, message to the packet's Source Address. The Pointer field in the ICMP message **SHOULD** point at the Option Type field. Otherwise, for multicast addresses, the ICMP message **MUST NOT** be sent.
- o The data within the option cannot change en-route to the packet's final destination.

The Home Address option **MUST** be placed as follows:

- o After the routing header, if that header is present
- o Before the Fragment Header, if that header is present
- o Before the AH Header or ESP Header, if either one of those headers is present

For each IPv6 packet header, the Home Address Option **MUST NOT** appear more than once. However, an encapsulated packet [15] **MAY** contain a separate Home Address option associated with each encapsulating IP header.

The inclusion of a Home Address destination option in a packet affects the receiving node's processing of only this single packet. No state is created or modified in the receiving node as a result of receiving a Home Address option in a packet. In particular, the presence of a Home Address option in a received packet **MUST NOT** alter the contents of the receiver's Binding Cache and **MUST NOT** cause any changes in the routing of subsequent packets sent by this receiving node.

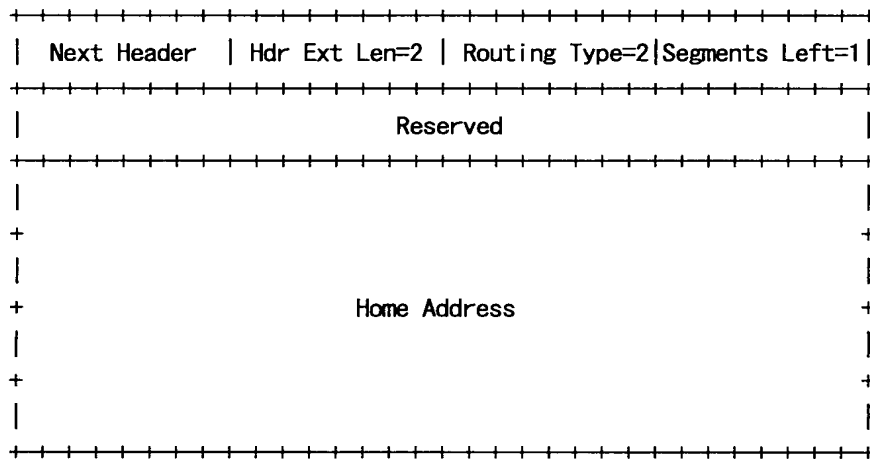
6.4 Type 2 Routing Header

Mobile IPv6 defines a new routing header variant, the type 2 routing header, to allow the packet to be routed directly from a correspondent to the mobile node's care-of address. The mobile node's care-of address is inserted into the IPv6 Destination Address field. Once the packet arrives at the care-of address, the mobile node retrieves its home address from the routing header, and this is used as the final destination address for the packet.

The new routing header uses a different type than defined for "regular" IPv6 source routing, enabling firewalls to apply different rules to source routed packets than to Mobile IPv6. This routing header type (type 2) is restricted to carry only one IPv6 address. All IPv6 nodes which process this routing header **MUST** verify that the address contained within is the node's own home address in order to prevent packets from being forwarded outside the node. The IP address contained in the routing header, since it is the mobile node's home address, **MUST** be a unicast routable address. Furthermore, if the scope of the home address is smaller than the scope of the care-of address, the mobile node **MUST** discard the packet (see Section 4.6).

6.4.1 Format

The type 2 routing header has the following format:



Next Header

8-bit selector. Identifies the type of header immediately following the routing header. Uses the same values as the IPv6 Next Header field [11].

Hdr Ext Len

2 (8-bit unsigned integer); length of the routing header in 8-octet units, not including the first 8 octets

Routing Type

2 (8-bit unsigned integer).

Segments Left

1 (8-bit unsigned integer).

Reserved

32-bit reserved field. The value **MUST** be initialized to zero by the sender, and **MUST** be ignored by the receiver.

Home Address

The Home Address of the destination Mobile Node.

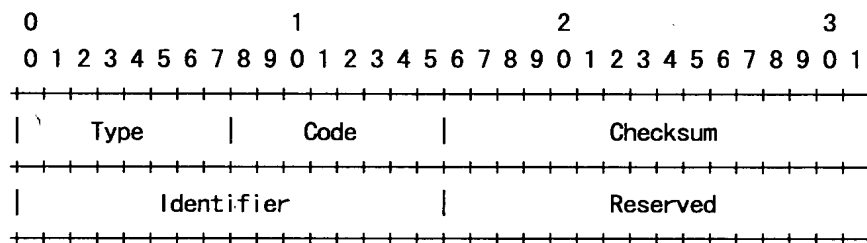
For a type 2 routing header, the Hdr Ext Len **MUST** be 2. The Segments Left value describes the number of route segments remaining; i.e., number of explicitly listed intermediate nodes still to be visited before reaching the final destination. Segments Left **MUST** be 1. The ordering rules for extension headers in an IPv6 packet are described in Section 4.1 of RFC 2460 [11]. The type 2 routing header defined

for Mobile IPv6 follows the same ordering as other routing headers. If both a type 0 and a type 2 routing header are present, the type 2 routing header should follow the other routing header. A packet containing such nested encapsulation should be created as if the inner (type 2) routing header was constructed first and then treated as an original packet by the outer (type 0) routing header construction process.

In addition, the general procedures defined by IPv6 for routing headers suggest that a received routing header MAY be automatically "reversed" to construct a routing header for use in any response packets sent by upper-layer protocols, if the received packet is authenticated [6]. This MUST NOT be done automatically for type 2 routing headers.

6.5 ICMP Home Agent Address Discovery Request Message

The ICMP Home Agent Address Discovery Request message is used by a mobile node to initiate the dynamic home agent address discovery mechanism, as described in Section 11.4.1. The mobile node sends the Home Agent Address Discovery Request message to the Mobile IPv6 Home-Agents anycast address [16] for its own home subnet prefix. (Note that the currently defined anycast addresses may not work with all prefix lengths other than those defined in RFC 2373 [3, 35].)



Type

150 <To Be Assigned by IANA>

Code

0

Checksum

The ICMP checksum [14].

Identifier

An identifier to aid in matching Home Agent Address Discovery

Reply messages to this Home Agent Address Discovery Request message.

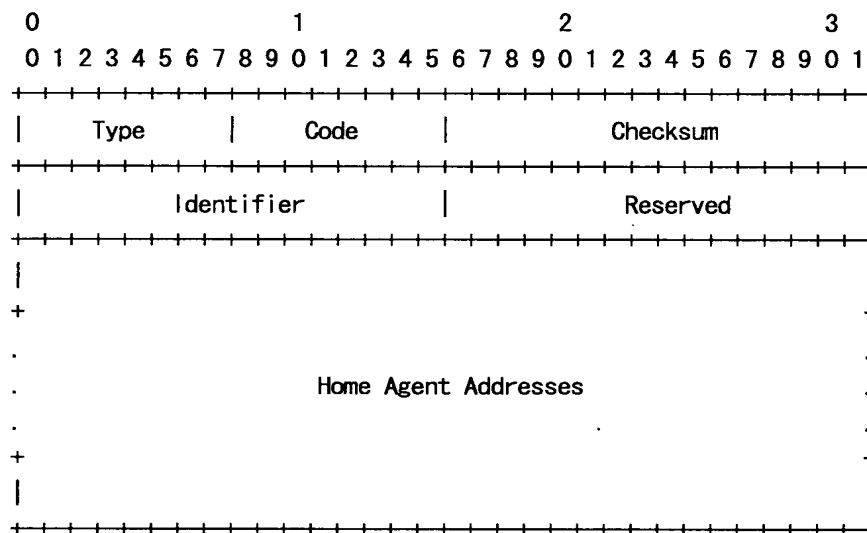
Reserved

This field is unused. It **MUST** be initialized to zero by the sender and **MUST** be ignored by the receiver.

The Source Address of the Home Agent Address Discovery Request message packet is typically one of the mobile node's current care-of addresses. At the time of performing this dynamic home agent address discovery procedure, it is likely that the mobile node is not registered with any home agent. Therefore, neither the nature of the address nor the identity of the mobile node can be established at this time. The home agent **MUST** then return the Home Agent Address Discovery Reply message directly to the Source Address chosen by the mobile node.

6.6 ICMP Home Agent Address Discovery Reply Message

The ICMP Home Agent Address Discovery Reply message is used by a home agent to respond to a mobile node that uses the dynamic home agent address discovery mechanism, as described in Section 10.5.



Type

151 <To Be Assigned by IANA>

Code

0

Checksum

The ICMP checksum [14].

Identifier

The identifier from the invoking Home Agent Address Discovery Request message.

Reserved

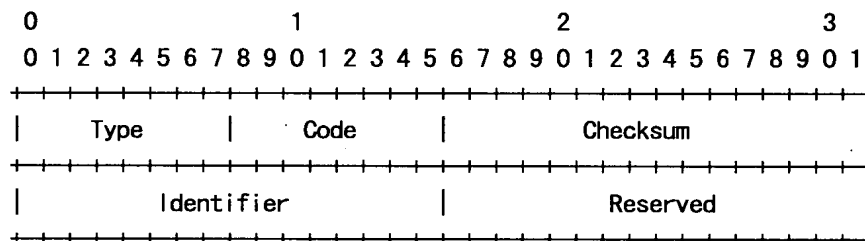
This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

Home Agent Addresses

A list of addresses of home agents on the home link for the mobile node. The number of addresses present in the list is indicated by the remaining length of the IPv6 packet carrying the Home Agent Address Discovery Reply message.

6.7 ICMP Mobile Prefix Solicitation Message Format

The ICMP Mobile Prefix Solicitation Message is sent by a mobile node to its home agent while it is away from home. The purpose of the message is to solicit a Mobile Prefix Advertisement from the home agent, which will allow the mobile node to gather prefix information about its home network. This information can be used to configure and update home address(es) according to changes in prefix information supplied by the home agent.



IP Fields:

Source Address

The mobile node's care-of address.

Destination Address

The address of the mobile node's home agent. This home agent must be on the link which the mobile node wishes to learn prefix information about.

Hop Limit

Set to an initial hop limit value, similarly to any other unicast packet sent by the mobile node.

Destination Option:

A Home Address destination option MUST be included.

ESP header:

IPsec headers MUST be supported and SHOULD be used as described in Section 5.4.

ICMP Fields:

Type

152 <To Be Assigned by IANA>

Code

0

Checksum

The ICMP checksum [14].

Identifier

An identifier to aid in matching a future Mobile Prefix Advertisement to this Mobile Prefix Solicitation.

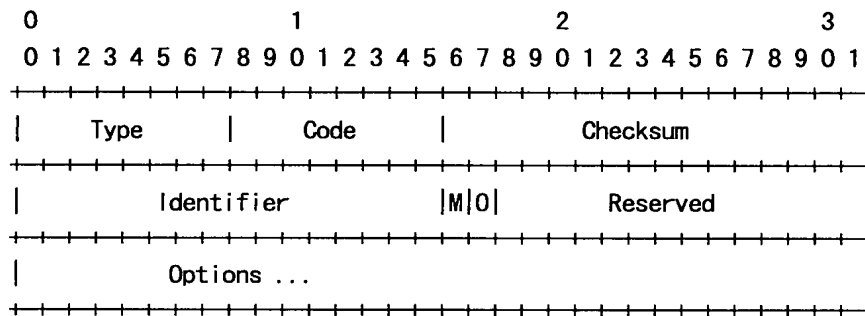
Reserved

This field is unused. It MUST be initialized to zero by the sender and MUST be ignored by the receiver.

The Mobile Prefix Solicitation messages may have options. These options **MUST** use the option format defined in RFC 2461 [12]. This document does not define any option types for the Mobile Prefix Solicitation message, but future documents may define new options. Home agents **MUST** silently ignore any options they do not recognize and continue processing the message.

6.8 ICMP Mobile Prefix Advertisement Message Format

A home agent will send a Mobile Prefix Advertisement to a mobile node to distribute prefix information about the home link while the mobile node is traveling away from the home network. This will occur in response to a Mobile Prefix Solicitation with an Advertisement, or by an unsolicited Advertisement sent according to the rules in Section 10.6.



IP Fields:

Source Address

The home agent's address as the mobile node would expect to see it (i.e., same network prefix).

Destination Address

If this message is a response to a Mobile Prefix Solicitation, this field contains the Source Address field from that packet. For unsolicited messages, the mobile node's care-of address **SHOULD** be used. Note that unsolicited messages can only be sent if the mobile node is currently registered with the home agent.

Routing header:

A type 2 routing header **MUST** be included.

ESP header:

IPsec headers **MUST** be supported and **SHOULD** be used as described in Section 5.4.

ICMP Fields:

Type

153 <To Be Assigned by IANA>

Code

0

Checksum

The ICMP checksum [14].

Identifier

An identifier to aid in matching this Mobile Prefix Advertisement to a previous Mobile Prefix Solicitation.

M

1-bit Managed Address Configuration flag. When set, hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. The use of this flag is described in [12, 13].

O

1-bit Other Stateful Configuration flag. When set, hosts use the administered (stateful) protocol for autoconfiguration of other (non-address) information. The use of this flag is described in [12, 13].

Reserved

This field is unused. It **MUST** be initialized to zero by the sender and **MUST** be ignored by the receiver.

The Mobile Prefix Advertisement messages may have options. These options **MUST** use the option format defined in RFC 2461 [12]. This document defines one option which may be carried in a Mobile Prefix Advertisement message, but future documents may define new options. Mobile nodes **MUST** silently ignore any options they do not recognize

and continue processing the message.

Prefix Information

Each message contains one or more Prefix Information options. Each option carries the prefix(es) that the mobile node should use to configure its home address(es). Section 10.6 describes which prefixes should be advertised to the mobile node.

The Prefix Information option is defined in Section 4.6.2 of RFC 2461 [12], with modifications defined in Section 7.2 of this specification. The home agent **MUST** use this modified Prefix Information option to send home network prefixes as defined in Section 10.6.1.

If the Advertisement is sent in response to a Mobile Prefix Solicitation, the home agent **MUST** copy the Identifier value from that message into the Identifier field of the Advertisement.

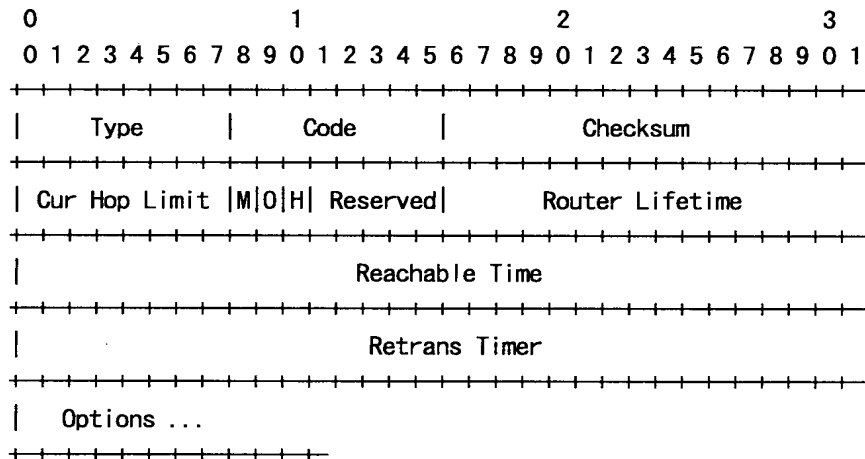
The home agent **MUST NOT** send more than one Mobile Prefix Advertisement message per second to any mobile node.

The M and O bits **MUST** be cleared if the Home Agent DHCPv6 support is not provided. If such support is provided then they are set in concert with the home network's administrative settings.

7. Modifications to IPv6 Neighbor Discovery

7.1 Modified Router Advertisement Message Format

Mobile IPv6 modifies the format of the Router Advertisement message [12] by the addition of a single flag bit to indicate that the router sending the Advertisement message is serving as a home agent on this link. The format of the Router Advertisement message is as follows:



This format represents the following changes over that originally specified for Neighbor Discovery [12]:

Home Agent (H)

The Home Agent (H) bit is set in a Router Advertisement to indicate that the router sending this Router Advertisement is also functioning as a Mobile IPv6 home agent on this link.

Reserved

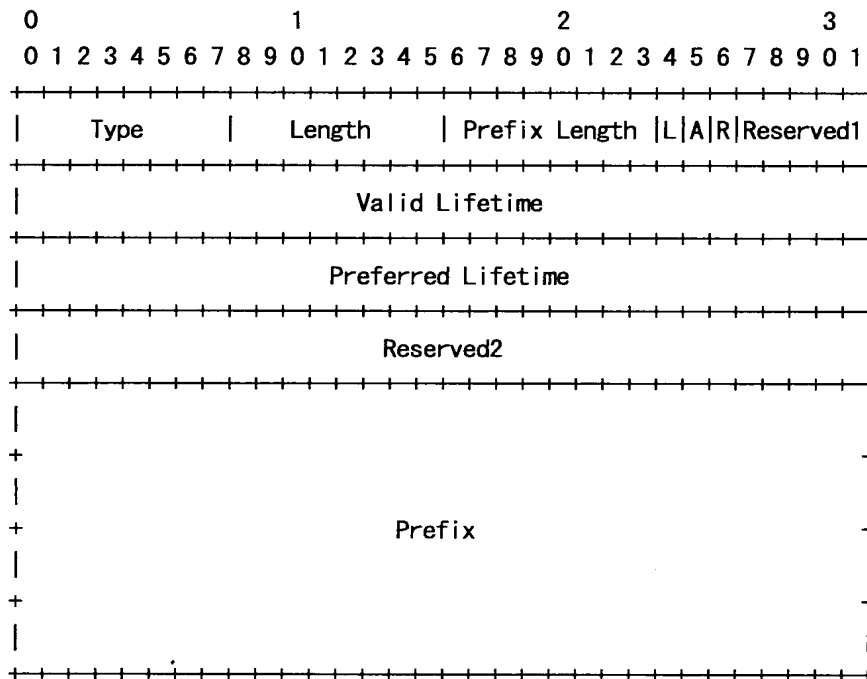
Reduced from a 6-bit field to a 5-bit field to account for the addition of the above bit.

7.2 Modified Prefix Information Option Format

Mobile IPv6 requires knowledge of a router's global address in building a Home Agents List as part of the dynamic home agent address discovery mechanism.

However, Neighbor Discovery [12] only advertises a router's link-local address, by requiring this address to be used as the IP Source Address of each Router Advertisement.

Mobile IPv6 extends Neighbor Discovery to allow a router to advertise its global address, by the addition of a single flag bit in the format of a Prefix Information option for use in Router Advertisement messages. The format of the Prefix Information option is as follows:



This format represents the following changes over that originally specified for Neighbor Discovery [12]:

Router Address (R)

1-bit router address flag. When set, indicates that the Prefix field contains a complete IP address assigned to the sending router. The indicated prefix is the first Prefix Length bits of the Prefix field. The router IP address has the same scope and conforms to the same lifetime values as the advertised prefix. This use of the Prefix field is compatible with its use in advertising the prefix itself, since Prefix Advertisement uses only the leading bits. Interpretation of this flag bit is thus independent of the processing required for the On-Link (L) and Autonomous Address-Configuration (A) flag bits.

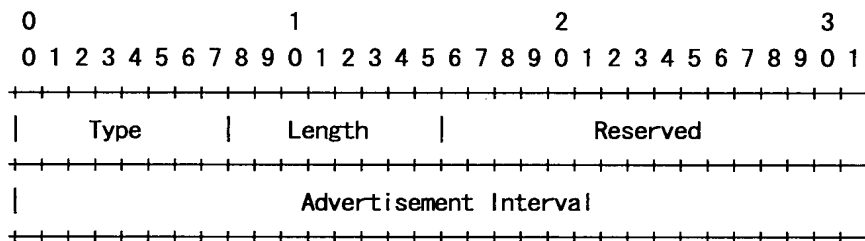
Reserved1

Reduced from a 6-bit field to a 5-bit field to account for the addition of the above bit.

In a Router Advertisement, a home agent **MUST**, and all other routers **MAY**, include at least one Prefix Information option with the Router

In addition, the following requirement can assist mobile nodes in movement detection. Barring changes in the prefixes for the link, routers that send multiple Router Advertisements with the Router Address (R) bit set in some of the included Prefix Information options SHOULD provide at least one option and router address which stays the same in all of the Advertisements.

Mobile IPv6 defines a new Advertisement Interval option, used in Router Advertisement messages to advertise the interval at which the sending router sends unsolicited multicast Router Advertisements. The format of the Advertisement Interval option is as follows:



7

8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 octets. The value of this field **MUST** be 1.

This field is unused. It **MUST** be initialized to zero by the sender and **MUST** be ignored by the receiver.

Advertisement Interval

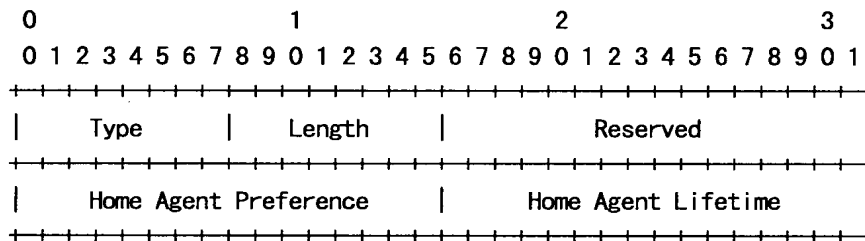
32-bit unsigned integer. The maximum time, in milliseconds, between successive unsolicited router Router Advertisement messages sent by this router on this network interface. Using the conceptual router configuration variables defined by Neighbor Discovery [12], this field **MUST** be equal to the value `MaxRtrAdvInterval`, expressed in milliseconds.

Routers **MAY** include this option in their Router Advertisements. A mobile node receiving a Router Advertisement containing this option **SHOULD** utilize the specified Advertisement Interval for that router in its movement detection algorithm, as described in Section 11.5.1.

This option **MUST** be silently ignored for other Neighbor Discovery messages.

7.4 New Home Agent Information Option Format

Mobile IPv6 defines a new Home Agent Information option, used in Router Advertisements sent by a home agent to advertise information specific to this router's functionality as a home agent. The format of the Home Agent Information option is as follows:



Type

8

Length

8-bit unsigned integer. The length of the option (including the type and length fields) in units of 8 octets. The value of this field **MUST** be 1.

Reserved

This field is unused. It **MUST** be initialized to zero by the sender and **MUST** be ignored by the receiver.

Home Agent Preference

16-bit unsigned integer. The preference for the home agent sending this Router Advertisement, for use in ordering the addresses returned to a mobile node in the Home Agent Addresses field of a Home Agent Address Discovery Reply message. Higher values mean more preferable. If this option is not included in a Router Advertisement in which the Home Agent (H) bit is set, the preference value for this home agent **MUST** be considered to be 0. Greater values indicate a more preferable home agent than lower values.

The manual configuration of the Home Agent Preference value is described in Section 8.4. In addition, the sending home agent **MAY** dynamically set the Home Agent Preference value, for example basing it on the number of mobile nodes it is currently serving or on its remaining resources for serving additional mobile nodes; such dynamic settings are beyond the scope of this document. Any such dynamic setting of the Home Agent Preference, however, **MUST** set the preference appropriately, relative to the default Home Agent Preference value of 0 that may be in use by some home agents on this link (i.e., a home agent not including a Home Agent Information option in its Router Advertisements will be considered to have a Home Agent Preference value of 0).

Home Agent Lifetime

16-bit unsigned integer. The lifetime associated with the home agent in units of seconds. The default value is the same as the Router Lifetime, as specified in the main body of the Router Advertisement. The maximum value corresponds to 18.2 hours. A value of 0 **MUST NOT** be used. The Home Agent Lifetime applies only to this router's usefulness as a home agent; it does not apply to information contained in other message fields or options.

Home agents **MAY** include this option in their Router Advertisements. This option **MUST NOT** be included in a Router Advertisement in which the Home Agent (H) bit (see Section 7.1) is not set. If this option is not included in a Router Advertisement in which the Home Agent (H) bit is set, the lifetime for this home agent **MUST** be considered to be the same as the Router Lifetime in the Router Advertisement. If multiple Advertisements are being sent instead of a single larger unsolicited multicast Advertisement, all of the multiple Advertisements with the Router Address (R) bit set **MUST** include this option with the same contents, otherwise this option **MUST** be omitted from all Advertisements.

This option **MUST** be silently ignored for other Neighbor Discovery

messages.

If both the Home Agent Preference and Home Agent Lifetime are set to their default values specified above, this option **SHOULD NOT** be included in the Router Advertisement messages sent by this home agent.

7.5 Changes to Sending Router Advertisements

The Neighbor Discovery protocol specification [12] limits routers to a minimum interval of 3 seconds between sending unsolicited multicast Router Advertisement messages from any given network interface (limited by MinRtrAdvInterval and MaxRtrAdvInterval), stating that:

"Routers generate Router Advertisements frequently enough that hosts will learn of their presence within a few minutes, but not frequently enough to rely on an absence of advertisements to detect router failure; a separate Neighbor Unreachability Detection algorithm provides failure detection."

This limitation, however, is not suitable to providing timely movement detection for mobile nodes. Mobile nodes detect their own movement by learning the presence of new routers as the mobile node moves into wireless transmission range of them (or physically connects to a new wired network), and by learning that previous routers are no longer reachable. Mobile nodes **MUST** be able to quickly detect when they move to a link served by a new router, so that they can acquire a new care-of address and send Binding Updates to register this care-of address with their home agent and to notify correspondent nodes as needed.

One method which can provide for faster movement detection, is to increase the rate at which unsolicited Router Advertisements are sent. Mobile IPv6 relaxes this limit such that routers **MAY** send unsolicited multicast Router Advertisements more frequently. This method can be applied where the router is expecting to provide service to visiting mobile nodes (e.g., wireless network interfaces), or on which it is serving as a home agent to one or more mobile nodes (who may return home and need to hear its Advertisements).

Routers supporting mobility **SHOULD** be able to be configured with a smaller MinRtrAdvInterval value and MaxRtrAdvInterval value to allow sending of unsolicited multicast Router Advertisements more often. The minimum allowed values are:

- o MinRtrAdvInterval 0.03 seconds
- o MaxRtrAdvInterval 0.07 seconds

In the case where the minimum intervals and delays are used, the mean time between unsolicited multicast router advertisements is 50ms. Use of these modified limits **MUST** be configurable (see also the configuration variable `MinDelayBetweenRas` in Section 13 which may also have to be modified accordingly). Systems where these values are available **MUST NOT** default to them, and **SHOULD** default to values specified in RFC 2461. Knowledge of the type of network interface and operating environment **SHOULD** be taken into account in configuring these limits for each network interface. This is important with some wireless links, where increasing the frequency of multicast beacons can cause considerable overhead. Routers **SHOULD** adhere to the intervals specified in RFC 2461 [12], if this overhead is likely to cause service degradation.

Additionally, the possible low values of `MaxRtrAdvInterval` may cause some problems with movement detection in some mobile nodes. To ensure that this is not a problem, Routers **SHOULD** add 20ms to any Advertisement Intervals sent in RAs, which are below 200 ms, in order to account for scheduling granularities on both the MN and the Router.

Note that multicast Router Advertisements are not always required in certain wireless networks that have limited bandwidth. Mobility detection or link changes in such networks may be done at lower layers. Router advertisements in such networks **SHOULD** be sent only when solicited. In such networks it **SHOULD** be possible to disable unsolicited multicast Router Advertisements on specific interfaces. The `MinRtrAdvInterval` and `MaxRtrAdvInterval` in such a case can be set to some high values.

Home agents **MUST** include the Source Link-Layer Address option in all Router Advertisements they send. This simplifies the process of returning home, as discussed in Section 11.5.4.

Note that according to RFC 2461 [12], `AdvDefaultLifetime` is by default based on the value of `MaxRtrAdvInterval`. `AdvDefaultLifetime` is used in the Router Lifetime field of Router Advertisements. Given that this field is expressed in seconds, a small `MaxRtrAdvInterval` value can result in a zero value for this field. To prevent this, routers **SHOULD** keep `AdvDefaultLifetime` in at least one second, even if the use of `MaxRtrAdvInterval` would result in a smaller value.

8. Requirements for Types of IPv6 Nodes

Mobile IPv6 places some special requirements on the functions provided by different types of IPv6 nodes. This section summarizes those requirements, identifying the functionality each requirement is intended to support.

The requirements are set for the following groups of nodes:

- o All IPv6 nodes.
- o All IPv6 nodes with support for route optimization.
- o All IPv6 routers.
- o All Mobile IPv6 home agents.
- o All Mobile IPv6 mobile nodes.

It is outside the scope of this specification to specify which of these groups are mandatory in IPv6. We only describe what is mandatory for a node that supports, for instance, route optimization. Other specifications are expected to define the extent of IPv6.

8.1 All IPv6 Nodes

Any IPv6 node may at any time be a correspondent node of a mobile node, either sending a packet to a mobile node or receiving a packet from a mobile node. There are no Mobile IPv6 specific MUST requirements for such nodes, and basic IPv6 techniques are sufficient. If a mobile node attempts to set up route optimization with a node with only basic IPv6 support, an ICMP error will signal that the node does not support such optimizations (Section 11.3.5), and communications will flow through the home agent.

An IPv6 node MUST NOT support the Home Address destination option, type 2 routing header, or the Mobility Header unless it fully supports the requirements listed in the next sections for either route optimization, mobile node, or home agent functionality.

8.2 IPv6 Nodes with Support for Route Optimization

Nodes that implement route optimization are a subset of all IPv6 nodes on the Internet. The ability of a correspondent node to participate in route optimization is essential for the efficient operation of the IPv6 Internet, for the following reasons:

- o Avoidance of congestion in the home network, and enabling the use

of lower-performance home agent equipment even for supporting thousands of mobile nodes.

- o Reduced network load across the entire Internet, as mobile devices begin to predominate.
- o Reduction of jitter and latency for the communications.
- o Greater likelihood of success for QoS signaling as tunneling is avoided and, again, fewer sources of congestion.
- o Improved robustness against network partitions, congestion, and other problems, since fewer routing path segments are traversed.

These effects combine to enable much better performance and robustness for communications between mobile nodes and IPv6 correspondent nodes. Route optimization introduces a small amount of additional state for the peers, some additional messaging, and up to 1.5 roundtrip delays before it can be turned on. However, it is believed that the benefits far outweigh the costs in most cases. Section 11.3.1 discusses how mobile nodes may avoid route optimization for some of the remaining cases, such as very short-term communications.

The following requirements apply to all correspondent nodes that support route optimization:

- o The node **MUST** be able validate a Home Address option using an existing Binding Cache entry, as described in Section 9.3.1.
- o The node **MUST** be able to insert a type 2 routing header into packets to be sent to a mobile node, as described in Section 9.3.2.
- o Unless the correspondent node is also acting as a mobile node, it **MUST** ignore type 2 routing headers and silently discard all packets that it has received with such headers.
- o The node **SHOULD** be able to interpret ICMP messages as described in Section 9.3.4.
- o The node **MUST** be able to send Binding Error messages as described in Section 9.3.3.
- o The node **MUST** be able to process Mobility Headers as described in Section 9.2.
- o The node **MUST** be able to participate in a return routability

procedure (Section 9.4).

- o The node **MUST** be able to process Binding Update messages (Section 9.5).
- o The node **MUST** be able to return a Binding Acknowledgement (Section 9.5.4).
- o The node **MUST** be able to maintain a Binding Cache of the bindings received in accepted Binding Updates, as described in Section 9.1 and Section 9.6.
- o The node **SHOULD** allow route optimization to be administratively enabled or disabled. The default **SHOULD** be enabled.

8.3 All IPv6 Routers

All IPv6 routers, even those not serving as a home agent for Mobile IPv6, have an effect on how well mobile nodes can communicate:

- o Every IPv6 router **SHOULD** be able to send an Advertisement Interval option (Section 7.3) in each of its Router Advertisements [12], to aid movement detection by mobile nodes (as in Section 11.5.1). The use of this option in Router Advertisements **SHOULD** be configurable.
- o Every IPv6 router **SHOULD** be able to support sending unsolicited multicast Router Advertisements at the faster rate described in Section 7.5. If the router supports a faster rate, the used rate **MUST** be configurable.
- o Each router **SHOULD** include at least one prefix with the Router Address (R) bit set and with its full IP address in its Router Advertisements (as described in Section 7.2).
- o Routers supporting filtering packets with routing headers **SHOULD** support different rules for type 0 and type 2 routing headers (see Section 6.4) so that filtering of source routed packets (type 0) will not necessarily limit Mobile IPv6 traffic which is delivered via type 2 routing headers.

8.4 IPv6 Home Agents

In order for a mobile node to operate correctly while away from home, at least one IPv6 router on the mobile node's home link must function as a home agent for the mobile node. The following additional

requirements apply to all IPv6 routers that serve as a home agent:

- o Every home agent **MUST** be able to maintain an entry in its Binding Cache for each mobile node for which it is serving as the home agent (Section 10.1 and Section 10.3.1).
- o Every home agent **MUST** be able to intercept packets (using proxy Neighbor Discovery [12]) addressed to a mobile node for which it is currently serving as the home agent, on that mobile node's home link, while the mobile node is away from home (Section 10.4.1).
- o Every home agent **MUST** be able to encapsulate [15] such intercepted packets in order to tunnel them to the primary care-of address for the mobile node indicated in its binding in the home agent's Binding Cache (Section 10.4.2).
- o Every home agent **MUST** support decapsulating [15] reverse tunneled packets sent to it from a mobile node's home address. Every home agent **MUST** also check that the source address in the tunneled packets corresponds to the currently registered location of the mobile node (Section 10.4.5).
- o The node **MUST** be able to process Mobility Headers as described in Section 10.2.
- o Every home agent **MUST** be able to return a Binding Acknowledgement in response to a Binding Update (Section 10.3.1).
- o Every home agent **MUST** maintain a separate Home Agents List for each link on which it is serving as a home agent, as described in Section 10.1 and Section 10.5.1.
- o Every home agent **MUST** be able to accept packets addressed to the Mobile IPv6 Home-Agents anycast address [16] for the subnet on which it is serving as a home agent, and **MUST** be able to participate in dynamic home agent address discovery (Section 10.5).
- o Every home agent **SHOULD** support a configuration mechanism to allow a system administrator to manually set the value to be sent by this home agent in the Home Agent Preference field of the Home Agent Information Option in Router Advertisements that it sends (Section 7.4).
- o Every home agent **SHOULD** support sending ICMP Mobile Prefix Advertisements (Section 6.8), and **SHOULD** respond to Mobile Prefix Solicitations (Section 6.7). If supported, this behavior **MUST** be configurable, so that home agents can be configured to avoid

sending such Prefix Advertisements according to the needs of the network administration in the home domain.

- o Every home agent **MUST** support IPsec ESP for protection of packets belonging to the return routability procedure (Section 10.4.6).
- o Every home agent **SHOULD** support the multicast group membership control protocols as described in Section 10.4.3. If this support is provided, the home agent **MUST** be capable of using it to determine which multicast data packets to forward via the tunnel to the mobile node.
- o Home agents **MAY** support stateful address autoconfiguration for mobile nodes as described in Section 10.4.4.

8.5 IPv6 Mobile Nodes

Finally, the following requirements apply to all IPv6 nodes capable of functioning as mobile nodes:

- o The node **MUST** maintain a Binding Update List (Section 11.1).
- o The node **MUST** support sending packets containing a Home Address option (Section 11.3.1), and follow the required IPsec interaction (Section 11.3.2).
- o The node **MUST** be able to perform IPv6 encapsulation and decapsulation [15].
- o The node **MUST** be able to process type 2 routing header as defined in Section 6.4 and Section 11.3.3.
- o The node **MUST** support receiving a Binding Error message (Section 11.3.6).
- o The node **MUST** support receiving ICMP errors (Section 11.3.5).
- o The node **MUST** support movement detection, care-of address formation, and returning home (Section 11.5).
- o The node **MUST** be able to process Mobility Headers as described in Section 11.2.
- o The node **MUST** support the return routability procedure (Section 11.6).
- o The node **MUST** be able to send Binding Updates, as specified in

Section 11.7.1 and Section 11.7.2.

- o The node **MUST** be able to receive and process Binding Acknowledgements, as specified in Section 11.7.3.
- o The node **MUST** support receiving a Binding Refresh Request (Section 6.1.2), by responding with a Binding Update.
- o The node **MUST** support receiving Mobile Prefix Advertisements (Section 11.4.3) and reconfiguring its home address based on the prefix information contained therein.
- o The node **SHOULD** support use of the dynamic home agent address discovery mechanism, as described in Section 11.4.1.
- o The node **MUST** allow route optimization to be administratively enabled or disabled. The default **SHOULD** be enabled.
- o The node **MAY** support the multicast address listener part of a multicast group membership protocol as described in Section 11.3.4. If this support is provided, the mobile node **MUST** be able to receive tunneled multicast packets from the home agent.
- o The node **MAY** support stateful address autoconfiguration mechanisms such as DHCPv6 [29] on the interface represented by the tunnel to the home agent.

9. Correspondent Node Operation

9.1 Conceptual Data Structures

IPv6 nodes with route optimization support maintain a Binding Cache of bindings for other nodes. A separate Binding Cache SHOULD be maintained by each IPv6 node for each of its unicast routable addresses. The Binding Cache MAY be implemented in any manner consistent with the external behavior described in this document, for example by being combined with the node's Destination Cache as maintained by Neighbor Discovery [12]. When sending a packet, the Binding Cache is searched before the Neighbor Discovery conceptual Destination Cache [12].

Each Binding Cache entry conceptually contains the following fields:

- o The home address of the mobile node for which this is the Binding Cache entry. This field is used as the key for searching the Binding Cache for the destination address of a packet being sent.
- o The care-of address for the mobile node indicated by the home address field in this Binding Cache entry.
- o A lifetime value, indicating the remaining lifetime for this Binding Cache entry. The lifetime value is initialized from the Lifetime field in the Binding Update that created or last modified this Binding Cache entry.
- o A flag indicating whether or not this Binding Cache entry is a home registration entry (applicable only on nodes which support home agent functionality).
- o The maximum value of the Sequence Number field received in previous Binding Updates for this home address. The Sequence Number field is 16 bits long. Sequence Number values MUST be compared modulo 2^{16} as explained in Section 9.5.1.
- o Usage information for this Binding Cache entry. This is needed to implement the cache replacement policy in use in the Binding Cache. Recent use of a cache entry also serves as an indication that a Binding Refresh Request should be sent when the lifetime of this entry nears expiration.

Binding Cache entries not marked as home registrations MAY be replaced at any time by any reasonable local cache replacement policy but SHOULD NOT be unnecessarily deleted. The Binding Cache for any one of a node's IPv6 addresses may contain at most one entry for each mobile node home address. The contents of a node's Binding Cache

MUST NOT be changed in response to a Home Address option in a received packet.

9.2 Processing Mobility Headers

Mobility Header processing MUST observe the following rules:

- o The checksum must be verified as per Section 6.1. Otherwise, the node MUST silently discard the message.
- o The MH Type field MUST have a known value (Section 6.1.1). Otherwise, the node MUST discard the message and issue a Binding Error message as described in Section 9.3.3, with Status field set to 2 (unrecognized MH Type value).
- o The Payload Proto field MUST be IPPROTO_NONE (59 decimal). Otherwise, the node MUST discard the message and SHOULD send ICMP Parameter Problem, Code 0, directly to the Source Address of the packet as specified in RFC 2463 [14]. Thus no Binding Cache information is used in sending the ICMP message. The Pointer field in the ICMP message SHOULD point at the Payload Proto field.
- o The Header Len field in the Mobility Header MUST NOT be less than the length specified for this particular type of message in Section 6.1. Otherwise, the node MUST discard the message and SHOULD send ICMP Parameter Problem, Code 0, directly to the Source Address of the packet as specified in RFC 2463 [14]. (The Binding Cache information is again not used.) The Pointer field in the ICMP message SHOULD point at the Header Len field.

Subsequent checks depend on the particular Mobility Header.

9.3 Packet Processing

This section describes how the correspondent node sends packets to the mobile node, and receives packets from it.

9.3.1 Receiving Packets with Home Address Option

Packets containing a Home Address option MUST be dropped if the given home address is not a unicast routable address.

Mobile nodes can include a Home Address destination option in a packet if they believe the correspondent node has a Binding Cache entry for the home address of a mobile node. Packets containing a Home Address option MUST be dropped if there is no corresponding Binding Cache entry. A corresponding Binding Cache entry MUST have the same home address as appears in the Home Address destination

option, and the currently registered care-of address MUST be equal to the source address of the packet. These tests MUST NOT be done for packets that contain a Home Address option and a Binding Update.

If the packet is dropped due the above tests, the correspondent node MUST send the Binding Error message as described in Section 9.3.3. The Status field in this message should be set to 1 (unknown binding for Home Address destination option).

The correspondent node MUST process the option in a manner consistent with exchanging the Home Address field from the Home Address option into the IPv6 header and replacing the original value of the Source Address field there. After all IPv6 options have been processed, it MUST be possible for upper layers to process the packet without the knowledge that it came originally from a care-of address or that a Home Address option was used.

The use of IPsec Authentication Header (AH) for the Home Address option is not required, except that if the IPv6 header of a packet is covered by AH, then the authentication MUST also cover the Home Address option; this coverage is achieved automatically by the definition of the Option Type code for the Home Address option, since it indicates that the data within the option cannot change en-route to the packet's final destination, and thus the option is included in the AH computation. By requiring that any authentication of the IPv6 header also cover the Home Address option, the security of the Source Address field in the IPv6 header is not compromised by the presence of a Home Address option.

When attempting to verify AH authentication data in a packet that contains a Home Address option, the receiving node MUST calculate the AH authentication data as if the following were true: The Home Address option contains the care-of address, and the source IPv6 address field of the IPv6 header contains the home address. This conforms with the calculation specified in Section 11.3.2.

9.3.2 Sending Packets to a Mobile Node

Before sending any packet, the sending node SHOULD examine its Binding Cache for an entry for the destination address to which the packet is being sent. If the sending node has a Binding Cache entry for this address, the sending node SHOULD use a type 2 routing header to route the packet to this mobile node (the destination node) by way of its care-of address. However, the mobile node MUST not do this in the following cases:

- o When sending an IPv6 Neighbor Discovery [12] packet.

- o Where otherwise noted in Section 6.1.

When calculating authentication data in a packet that contains a type 2 routing header, the correspondent node MUST calculate the AH authentication data as if the following were true: The routing header contains the care-of address, the destination IPv6 address field of the IPv6 header contains the home address, and the Segments Left field is zero. The IPsec Security Policy Database lookup MUST be based on the mobile node's home address.

For instance, assuming there are no additional routing headers in this packet beyond those needed by Mobile IPv6, the correspondent node could set the fields in the packet's IPv6 header and routing header as follows:

- o The Destination Address in the packet's IPv6 header is set to the mobile node's home address (the original destination address to which the packet was being sent).
- o The routing header is initialized to contain a single route segment, containing the mobile node's care-of address copied from the Binding Cache entry. The Segments Left field is, however, temporarily set to zero.

The IP layer will insert the routing header before performing any necessary IPsec processing. Once all IPsec processing has been performed, the node swaps the IPv6 destination field with the Home Address field in the routing header, sets the Segments Left field to one, and sends the packet. This ensures the AH calculation is done on the packet in the form it will have on the receiver after advancing the routing header.

Following the definition of a type 2 routing header in Section 6.4, this packet will be routed to the mobile node's care-of address, where it will be delivered to the mobile node (the mobile node has associated the care-of address with its network interface).

Note that following the above conceptual model in an implementation creates some additional requirements for path MTU discovery since the layer that decides the packet size (e.g., TCP and applications using UDP) needs to be aware of the size of the headers added by the IP layer on the sending node.

If, instead, the sending node has no Binding Cache entry for the destination address to which the packet is being sent, the sending node simply sends the packet normally, with no routing header. If the destination node is not a mobile node (or is a mobile node that is currently at home), the packet will be delivered directly to this

node and processed normally by it. If, however, the destination node is a mobile node that is currently away from home, the packet will be intercepted by the mobile node's home agent and tunneled to the mobile node's current primary care-of address.

9.3.3 Sending Binding Error Messages

Section 9.2 and Section 9.3.1 describe error conditions that lead to a need to send a Binding Error message.

A Binding Error message is sent directly to the address that appeared in the IPv6 Source Address field of the offending packet. If the Source Address field does not contain a unicast address, the Binding Error message **MUST NOT** be sent.

The Home Address field in the Binding Error message **MUST** be copied from the Home Address field in the Home Address destination option of the offending packet, or set to the unspecified address if no such option appeared in the packet.

Note that the IPv6 Source Address and Home Address field values discussed above are the values from the wire, i.e., before any modifications possibly performed as specified in Section 9.3.1.

Binding Error messages **SHOULD** be subject to rate limiting in the same manner as is done for ICMPv6 messages [14].

9.3.4 Receiving ICMP Error Messages

When the correspondent node has a Binding Cache entry for a mobile node, all traffic destined to the mobile node goes directly to the current care-of address of the mobile node using a routing header. Any ICMP error message caused by packets on their way to the care-of address will be returned in the normal manner to the correspondent node.

On the other hand, if the correspondent node has no Binding Cache entry for the mobile node, the packet will be routed through the mobile node's home link. Any ICMP error message caused by the packet on its way to the mobile node while in the tunnel, will be transmitted to the mobile node's home agent. By the definition of IPv6 encapsulation [15], the home agent **MUST** relay certain ICMP error messages back to the original sender of the packet, which in this case is the correspondent node.

Thus, in all cases, any meaningful ICMP error messages caused by packets from a correspondent node to a mobile node will be returned to the correspondent node. If the correspondent node receives

persistent ICMP Destination Unreachable messages after sending packets to a mobile node based on an entry in its Binding Cache, the correspondent node SHOULD delete this Binding Cache entry. Note that if the mobile node continues to send packets with the Home Address destination option to this correspondent node, they will be dropped due to the lack of a binding. For this reason it is important that only persistent ICMP messages lead to the deletion of the Binding Cache entry.

9.4 Return Routability Procedure

This subsection specifies actions taken by a correspondent node during the return routability procedure.

9.4.1 Receiving Home Test Init Messages

Upon receiving a Home Test Init message, the correspondent node verifies the following:

- o The packet MUST NOT include a Home Address destination option.

Any packet carrying a Home Test Init message which fails to satisfy all of these tests MUST be silently ignored.

Otherwise, in preparation for sending the corresponding Home Test Message, the correspondent node checks that it has the necessary material to engage in a return routability procedure, as specified in Section 5.2. The correspondent node MUST have a secret Kcn and a nonce. If it does not have this material yet, it MUST produce it before continuing with the return routability procedure.

Section 9.4.3 specifies further processing.

9.4.2 Receiving Care-of Test Init Messages

Upon receiving a Care-of Test Init message, the correspondent node verifies the following:

- o The packet MUST NOT include a Home Address destination option.

Any packet carrying a Care-of Test Init message which fails to satisfy all of these tests MUST be silently ignored.

Otherwise, in preparation for sending the corresponding Care-of Test Message, the correspondent node checks that it has the necessary material to engage in a return routability procedure in the manner described in Section 9.4.1.

Section 9.4.4 specifies further processing.

9.4.3 Sending Home Test Messages

The correspondent node creates a home keygen token and uses the current nonce index as the Home Nonce Index. It then creates a Home Test message (Section 6.1.5) and sends it to the mobile node at the latter's home address.

9.4.4 Sending Care-of Test Messages

The correspondent node creates a care-of nonce and uses the current nonce index as the Care-of Nonce Index. It then creates a Care-of Test message (Section 6.1.6) and sends it to the mobile node at the latter's care-of address.

9.5 Processing Bindings

This section explains how the correspondent node processes messages related to bindings. These messages are:

- o Binding Update
- o Binding Refresh Request
- o Binding Acknowledgement
- o Binding Error

9.5.1 Receiving Binding Updates

Before accepting a Binding Update, the receiving node **MUST** validate the Binding Update according to the following tests:

- o The packet **MUST** contain a unicast routable home address, either in the Home Address option or in the Source Address, if the Home Address option is not present.
- o The Sequence Number field in the Binding Update is greater than the Sequence Number received in the previous valid Binding Update for this home address, if any.

If the receiving node has no Binding Cache entry for the indicated home address, it **MUST** accept any Sequence Number value in a received Binding Update from this mobile node.

This Sequence Number comparison **MUST** be performed modulo 2^{16} ,

i.e., the number is a free running counter represented modulo 65536. A Sequence Number in a received Binding Update is considered less than or equal to the last received number if its value lies in the range of the last received number and the preceding 32768 values, inclusive. For example, if the last received sequence number was 15, then messages with sequence numbers 0 through 15, as well as 32783 through 65535, would be considered less than or equal.

When the Home Registration (H) bit is not set, the following are also required:

- o A Nonce Indices mobility option **MUST** be present, and the Home and Care-of Nonce Index values in this option **MUST** be recent enough to be recognized by the correspondent node. (Care-of Nonce Index values are not inspected for requests to delete a binding.)
- o The correspondent node **MUST** re-generate the home keygen token and the care-of keygen token from the information contained in the packet. It then generates the binding management key Kbm and uses it to verify the authenticator field in the Binding Update as specified in Section 6.1.7.
- o The Binding Authorization Data mobility option **MUST** be present, and its contents **MUST** satisfy rules presented in Section 5.2.6. Note that a care-of address different from the Source Address **MAY** have been specified by including an Alternate Care-of Address mobility option in the Binding Update. When such a message is received and the return routability procedure is used as an authorization method, the correspondent node **MUST** verify the authenticator by using the address within the Alternate Care-of Address in the calculations.
- o The Binding Authorization Data mobility option **MUST** be the last option and **MUST NOT** have trailing padding.

If the Home Registration (H) bit is set, the Nonce Indices mobility option **MUST NOT** be present.

If the mobile node sends a sequence number which is not greater than the sequence number from the last valid Binding Update for this home address, then the receiving node **MUST** send back a Binding Acknowledgement with status code 135, and the last accepted sequence number in the Sequence Number field of the Binding Acknowledgement.

If a binding already exists for the given home address and the home registration flag has a different value than the Home Registration (H) bit in the Binding Update, then the receiving node **MUST** send back

a Binding Acknowledgement with status code 139 (registration type change disallowed). The home registration flag stored in the Binding Cache entry **MUST NOT** be changed.

If the receiving node no longer recognizes the Home Nonce Index value, Care-of Nonce Index value, or both values from the Binding Update, then the receiving node **MUST** send back a Binding Acknowledgement with status code 136, 137, or 138, respectively.

For packets carrying Binding Updates that fail to satisfy all of these tests for any reason other than insufficiency of the Sequence Number, registration type change, or expired nonce index values, they **MUST** be silently discarded.

If the Binding Update is valid according to the tests above, then the Binding Update is processed further as follows:

- o The Sequence Number value received from a mobile node in a Binding Update is stored by the receiving node in its Binding Cache entry for the given home address.
- o If the Lifetime specified in the Binding Update is nonzero and the specified care-of address is not equal to the home address for the binding, then this is a request to cache a binding for the home address. If the Home Registration (H) bit is set in the Binding Update, the Binding Update is processed according to the procedure specified in Section 10.3.1; otherwise, it is processed according to the procedure specified in Section 9.5.2.
- o If the Lifetime specified in the Binding Update is zero or the specified care-of address matches the home address for the binding, then this is a request to delete the cached binding for the home address. In this case, the Binding Update **MUST** include a valid home nonce index, and the care-of nonce index **MUST** be ignored by the correspondent node. The generation of the binding management key depends then exclusively on the home keygen token (Section 5.2.5). If the Home Registration (H) bit is set in the Binding Update, the Binding Update is processed according to the procedure specified in Section 10.3.2; otherwise, it is processed according to the procedure specified in Section 9.5.3.

The specified care-of address **MUST** be determined as follows:

- o If the Alternate Care-of Address option is present, the care-of address is the address in that option.
- o Otherwise, the care-of address is the Source Address field in the packet's IPv6 header.

The home address for the binding **MUST** be determined as follows:

- o If the Home Address destination option is present, the home address is the address in that option.
- o Otherwise, the home address is the Source Address field in the packet's IPv6 header.

9.5.2 Requests to Cache a Binding

This section describes the processing of a valid Binding Update that requests a node to cache a binding, for which the Home Registration (H) bit is not set in the Binding Update.

In this case, the receiving node **SHOULD** create a new entry in its Binding Cache for this home address, or update its existing Binding Cache entry for this home address, if such an entry already exists. The lifetime for the Binding Cache entry is initialized from the Lifetime field specified in the Binding Update, although this lifetime **MAY** be reduced by the node caching the binding; the lifetime for the Binding Cache entry **MUST NOT** be greater than the Lifetime value specified in the Binding Update. Any Binding Cache entry **MUST** be deleted after the expiration of its lifetime.

Note that if the mobile node did not request a Binding Acknowledgement, it is not aware of the selected shorter lifetime. The mobile node may thus use route optimization and send packets with the Home Address destination option. As discussed in Section 9.3.1, such packets will be dropped if there is no binding. This situation is recoverable, but can cause temporary packet loss.

The correspondent node **MAY** refuse to accept a new Binding Cache entry, if it does not have sufficient resources. A new entry **MAY** also be refused if the correspondent node believes its resources are utilized more efficiently in some other purpose, such as serving another mobile node with higher amount of traffic. In both cases the correspondent node **SHOULD** return a Binding Acknowledgement with status value 130.

9.5.3 Requests to Delete a Binding

This section describes the processing of a valid Binding Update that requests a node to delete a binding, when the Home Registration (H) bit is not set in the Binding Update.

Any existing binding for the given home address **MUST** be deleted. A Binding Cache entry for the home address **MUST NOT** be created in

response to receiving the Binding Update.

If the Binding Cache entry was created by use of return routability nonces, the correspondent node **MUST** ensure that the same nonces are not used again with the particular home and care-of address. If both nonces are still valid, the correspondent node has to remember the particular combination of nonce indexes, addresses, and sequence number as illegal, until at least one of the nonces has become too old.

9.5.4 Sending Binding Acknowledgements

A Binding Acknowledgement may be sent to indicate receipt of a Binding Update as follows:

- o If the Binding Update was discarded as described in Section 9.2 or Section 9.5.1, a Binding Acknowledgement **MUST NOT** be sent. Otherwise the treatment depends on the below rules.
- o If the Acknowledge (A) bit set is set in the Binding Update, a Binding Acknowledgement **MUST** be sent. Otherwise, the treatment depends on the below rule.
- o If the node rejects the Binding Update due to an expired nonce index, sequence number being out of window (Section 9.5.1), or insufficiency of resources (Section 9.5.2), a Binding Acknowledgement **MUST** be sent. If the node accepts the Binding Update, the Binding Acknowledgement **SHOULD NOT** be sent.

If the node accepts the Binding Update and creates or updates an entry for this binding, the Status field in the Binding Acknowledgement **MUST** be set to a value less than 128. Otherwise, the Status field **MUST** be set to a value greater than or equal to 128. Values for the Status field are described in Section 6.1.8 and in the IANA registry of assigned numbers [19].

If the Status field in the Binding Acknowledgement contains the value 136 (expired home nonce index), 137 (expired care-of nonce index), or 138 (expired nonces) then the message **MUST NOT** include the Binding Authorization Data mobility option. Otherwise, the Binding Authorization Data mobility option **MUST** be included, and **MUST** meet the specific authentication requirements for Binding Acknowledgements as defined in Section 5.2.

If the Source Address field of the IPv6 header that carried the Binding Update does not contain a unicast address, the Binding Acknowledgement **MUST NOT** be sent, and the Binding Update packet **MUST** be silently discarded. Otherwise, the acknowledgement **MUST** be sent

to the Source Address. Unlike the treatment of regular packets, this addressing procedure does not use information from the Binding Cache. However, a routing header is needed in some cases. If the Source Address is the home address of the mobile node, i.e., the Binding Update did not contain a Home Address destination option, then the Binding Acknowledgement **MUST** be sent to that address, and the routing header **MUST NOT** be used. Otherwise, the Binding Acknowledgement **MUST** be sent using a type 2 routing header which contains the mobile node's home address.

9.5.5 Sending Binding Refresh Requests

If a Binding Cache entry being deleted is still in active use in sending packets to a mobile node, the next packet sent to the mobile node will be routed normally to the mobile node's home link. Communication with the mobile node continues, but the tunneling from the home network creates additional overhead and latency in delivering packets to the mobile node.

If the sender knows that the Binding Cache entry is still in active use, it **MAY** send a Binding Refresh Request message to the mobile node in an attempt to avoid this overhead and latency due to deleting and recreating the Binding Cache entry.

The correspondent node **MAY** retransmit Binding Refresh Request messages provided that rate limitation is applied. The correspondent node **MUST** stop retransmitting when it receives a Binding Update.

9.6 Cache Replacement Policy

Conceptually, a node maintains a separate timer for each entry in its Binding Cache. When creating or updating a Binding Cache entry in response to a received and accepted Binding Update, the node sets the timer for this entry to the specified Lifetime period. Any entry in a node's Binding Cache **MUST** be deleted after the expiration of the Lifetime specified in the Binding Update from which the entry was created or last updated.

Each node's Binding Cache will, by necessity, have a finite size. A node **MAY** use any reasonable local policy for managing the space within its Binding Cache.

A node **MAY** choose to drop any entry already in its Binding Cache in order to make space for a new entry. For example, a "least-recently used" (LRU) strategy for cache entry replacement among entries is likely to work well unless the size of the Binding Cache is substantially insufficient. When entries are deleted, the correspondent node **MUST** follow the rules in Section 5.2.8 in order to

guard the return routability procedure against replay attacks.

If the node sends a packet to a destination for which it has dropped the entry from its Binding Cache, the packet will be routed through the mobile node's home link. The mobile node can detect this, and establish a new binding if necessary.

However, if the mobile node believes that the binding still exists, it may use route optimization and send packets with the Home Address destination option. This can create temporary packet loss, as discussed earlier in the context of binding lifetime reductions performed by the correspondent node (Section 9.5.2).

10. Home Agent Operation

10.1 Conceptual Data Structures

Each home agent **MUST** maintain a Binding Cache and Home Agents List.

The rules for maintaining a Binding Cache are the same for home agents and correspondent nodes, and have already been described in Section 9.1.

The Home Agents List is maintained by each home agent, recording information about each router on the same link which is acting as a home agent: this list is used by the dynamic home agent address discovery mechanism. A router is known to be acting as a home agent, if it sends a Router Advertisement in which the Home Agent (H) bit is set. When the lifetime for a list entry (defined below) expires, that entry is removed from the Home Agents List. The Home Agents List is thus similar to the Default Router List conceptual data structure maintained by each host for Neighbor Discovery [12]. The Home Agents List **MAY** be implemented in any manner consistent with the external behavior described in this document.

Each home agent maintains a separate Home Agents List for each link on which it is serving as a home agent. A new entry is created or an existing entry is updated in response to receipt of a valid Router Advertisement in which the Home Agent (H) bit is set. Each Home Agents List entry conceptually contains the following fields:

- o The link-local IP address of a home agent on the link. This address is learned through the Source Address of the Router Advertisements [12] received from the router.
- o One or more global IP addresses for this home agent. Global addresses are learned through Prefix Information options with the Router Address (R) bit set, received in Router Advertisements from this link-local address. Global addresses for the router in a Home Agents List entry **MUST** be deleted once the prefix associated with that address is no longer valid [12].
- o The remaining lifetime of this Home Agents List entry. If a Home Agent Information Option is present in a Router Advertisement received from a home agent, the lifetime of the Home Agents List entry representing that home agent is initialized from the Home Agent Lifetime field in the option (if present); otherwise, the lifetime is initialized from the Router Lifetime field in the received Router Advertisement. If Home Agents List entry lifetime reaches zero, the entry **MUST** be deleted from the Home Agents List.

- o The preference for this home agent: higher values indicate a more preferable home agent. The preference value is taken from the Home Agent Preference field in the received Router Advertisement, if the Router Advertisement contains a Home Agent Information Option, and is otherwise set to the default value of 0. A home agent uses this preference in ordering the Home Agents List when it sends an ICMP Home Agent Address Discovery message.

10.2 Processing Mobility Headers

All IPv6 home agents **MUST** observe the rules described in Section 9.2 when processing Mobility Headers.

10.3 Processing Bindings

10.3.1 Primary Care-of Address Registration

When a node receives a Binding Update, it **MUST** validate it and determine the type of Binding Update according to the steps described in Section 9.5.1. Furthermore, it **MUST** authenticate the Binding Update as described in Section 5.1. An authorization step specific for the home agent is also needed to ensure that only the right node can control a particular home address. This is provided through the home address unequivocally identifying the security association that must be used.

This section describes the processing of a valid and authorized Binding Update, when it requests the registration of the mobile node's primary care-of address.

To begin processing the Binding Update, the home agent **MUST** perform the following sequence of tests:

- o If the node implements only correspondent node functionality, or has not been configured to act as a home agent, then the node **MUST** reject the Binding Update. The node **MUST** then also return a Binding Acknowledgement to the mobile node, in which the Status field is set to 131 (home registration not supported).
- o Else, if the home address for the binding (the Home Address field in the packet's Home Address option) is not an on-link IPv6 address with respect to the home agent's current Prefix List, then the home agent **MUST** reject the Binding Update and **SHOULD** return a Binding Acknowledgement to the mobile node, in which the Status field is set to 132 (not home subnet).
- o Else, if the home agent chooses to reject the Binding Update for

any other reason (e.g., insufficient resources to serve another mobile node as a home agent), then the home agent SHOULD return a Binding Acknowledgement to the mobile node, in which the Status field is set to an appropriate value to indicate the reason for the rejection.

- o A Home Address destination option MUST be present in the message. It MUST be validated as described in Section 9.3.1 with the following additional rule. The Binding Cache entry existence test MUST NOT be done for IPsec packets when the Home Address option contains an address for which the receiving node could act as a home agent.

If home agent accepts the Binding Update, it MUST then create a new entry in its Binding Cache for this mobile node, or update its existing Binding Cache entry, if such an entry already exists. The Home Address field as received in the Home Address option provides the home address of the mobile node.

The home agent MUST mark this Binding Cache entry as a home registration to indicate that the node is serving as a home agent for this binding. Binding Cache entries marked as a home registration MUST be excluded from the normal cache replacement policy used for the Binding Cache (Section 9.6) and MUST NOT be removed from the Binding Cache until the expiration of the Lifetime period.

Unless this home agent already has a binding for the given home address, the home agent MUST perform Duplicate Address Detection [13] on the mobile node's home link before returning the Binding Acknowledgement. This ensures that no other node on the home link was using the mobile node's home address when the Binding Update arrived. If this Duplicate Address Detection fails for the given home address or an associated link local address, then the home agent MUST reject the complete Binding Update and MUST return a Binding Acknowledgement to the mobile node, in which the Status field is set to 134 (Duplicate Address Detection failed). When the home agent sends a successful Binding Acknowledgement to the mobile node, the home agent assures to the mobile node that its address(es) will continue to be kept unique by the home agent at least as long as the lifetime granted for the binding is not over.

The specific addresses which are to be tested before accepting the Binding Update, and later to be defended by performing Duplicate Address Detection, depend on the setting of the Link-Local Address Compatibility (L) bit, as follows:

- o L=0: Defend only the given address. Do not derive a link-local address.

- o L=1: Defend both the given non link-local unicast (home) address and the derived link-local. The link-local address is derived by replacing the subnet prefix in the mobile node's home address with the link-local prefix.

The lifetime of the Binding Cache entry depends on a number of factors:

- o The lifetime for the Binding Cache entry MUST NOT be greater than the Lifetime value specified in the Binding Update.
- o The lifetime for the Binding Cache entry MUST NOT be greater than the remaining valid lifetime for the subnet prefix in the mobile node's home address specified with the Binding Update. The remaining valid lifetime for this prefix is determined by the home agent based on its own Prefix List entry for this prefix [12].

The remaining preferred lifetime SHOULD NOT have any impact on the lifetime for the binding cache entry.

The home agent MUST remove a binding when the valid lifetime of the prefix associated with it expires.

- o The home agent MAY further decrease the specified lifetime for the binding, for example based on a local policy. The resulting lifetime is stored by the home agent in the Binding Cache entry, and this Binding Cache entry MUST be deleted by the home agent after the expiration of this lifetime.

Regardless of the setting of the Acknowledge (A) bit in the Binding Update, the home agent MUST return a Binding Acknowledgement to the mobile node, constructed as follows:

- o The Status field MUST be set to a value indicating success. The value 1 (accepted but prefix discovery necessary) MUST be used if the subnet prefix of the specified home address is deprecated, becomes deprecated during the lifetime of the binding, or becomes invalid at the end of the lifetime. The value 0 MUST be used otherwise. For the purposes of comparing the binding and prefix lifetimes, the prefix lifetimes are first converted into units of four seconds by ignoring the two least significant bits.
- o The Key Management Mobility Capability (K) bit is set if the following conditions are all fulfilled, and cleared otherwise:
 - * The Key Management Mobility Capability (K) bit was set in the Binding Update.

- * The IPsec security associations between the mobile node and the home agent have been established dynamically.
- * The home agent has the capability to update its endpoint in the used key management protocol to the new care-of address every time it moves

Depending on the final value of the bit in the Binding Acknowledgement, the home agent SHOULD perform the following actions:

K = 0

Discard key management connections, if any, to the old care-of address. If the mobile node did not have a binding before sending this Binding Update, discard the connections to the home address.

K = 1

Move the peer endpoint of the key management protocol connection, if any, to the new care-of address. For an IKE phase 1 connection, this means that any IKE packets sent to the peer are sent to this address, and packets from this address with the original ISAKMP cookies are accepted.

Note that Section 2.5.3 in RFC 2408 [8] Section 2.5.3 states three specifies rules that ISAKMP cookies must satisfy: they must depend on specific parties and they can only have been generated by the entity itself. Then it recommends a particular way to do this, namely a hash of IP addresses. With the K bit set to 1, the recommended implementation technique does not work directly. To satisfy the two rules, the specific parties must be treated as the original IP addresses, not the ones in use at the specific moment.

- o The Sequence Number field MUST be copied from the Sequence Number given in the Binding Update.
- o The Lifetime field MUST be set to the remaining lifetime for the binding as set by the home agent in its home registration Binding Cache entry for the mobile node, as described above.
- o If the home agent stores the Binding Cache entry in nonvolatile storage, then the Binding Refresh Advice mobility option MUST be omitted. Otherwise, the home agent MAY include this option to suggest that the mobile node refreshes its binding sooner than the

actual lifetime of the binding ends.

If the Binding Refresh Advice mobility option is present, the Refresh Interval field in the option **MUST** be set to a value less than the Lifetime value being returned in the Binding Acknowledgement. This indicates that the mobile node **SHOULD** attempt to refresh its home registration at the indicated shorter interval. The home agent **MUST** still retain the registration for the Lifetime period, even if the mobile node does not refresh its registration within the Refresh period.

The rules for selecting the Destination IP address (and possibly routing header construction) for the Binding Acknowledgement to the mobile node are the same as in Section 9.5.4.

In addition, the home agent **MUST** follow the procedure defined in Section 10.4.1 to intercept packets on the mobile node's home link addressed to the mobile node, while the home agent is serving as the home agent for this mobile node. The home agent **MUST** also be prepared to accept reverse tunneled packets from the new care-of address of the mobile node, as described in Section 10.4.5. Finally, the home agent **MUST** also propagate new home network prefixes, as described in Section 10.6.

10.3.2 Primary Care-of Address De-Registration

A binding may need to be de-registered when the mobile node returns home, or when the mobile node knows that it will soon not have any care-of addresses in the visited network.

A Binding Update is validated and authorized in the manner described in the previous section. This section describes the processing of a valid Binding Update that requests the receiving node to no longer serve as its home agent, de-registering its primary care-of address.

To begin processing the Binding Update, the home agent **MUST** perform the following test:

- o If the receiving node has no entry marked as a home registration in its Binding Cache for this mobile node, then this node **MUST** reject the Binding Update and **SHOULD** return a Binding Acknowledgement to the mobile node, in which the Status field is set to 133 (not home agent for this mobile node).

If the home agent does not reject the Binding Update as described above, then it **MUST** delete any existing entry in its Binding Cache for this mobile node. Then, the home agent **MUST** return a Binding Acknowledgement to the mobile node, constructed as follows:

- o The Status field **MUST** be set to a value 0, indicating success.
- o The Key Management Mobility Capability (K) bit is set or cleared, and actions based on its value are performed as described in the previous section. The mobile node's home address is used as its new care-of address for the purposes of moving the key management connection to a new endpoint.
- o The Sequence Number field **MUST** be copied from the Sequence Number given in the Binding Update.
- o The Lifetime field **MUST** be set to zero.
- o The Binding Refresh Advice mobility option **MUST** be omitted.

In addition, the home agent **MUST** stop intercepting packets on the mobile node's home link that are addressed to the mobile node (Section 10.4.1).

The rules for selecting the Destination IP address (and, if required, routing header construction) for the Binding Acknowledgement to the mobile node are the same as in the previous section. When the Status field in the Binding Acknowledgement is greater than or equal to 128 and the Source Address of the Binding Update is on the home link, the home agent **MUST** send it to the mobile node's link layer address (retrieved either from the Binding Update or through Neighbor Solicitation).

10.4 Packet Processing

10.4.1 Intercepting Packets for a Mobile Node

While a node is serving as the home agent for mobile node it **MUST** attempt to intercept packets on the mobile node's home link that are addressed to the mobile node.

In order to do this, when a node begins serving as the home agent it **MUST** multicast onto the home link a Neighbor Advertisement message [12] on behalf of the mobile node. For the home address specified in the Binding Update, the home agent sends a Neighbor Advertisement message [12] to the all-nodes multicast address on the home link, to advertise the home agent's own link-layer address for this IP address on behalf of the mobile node. If the Link-Layer Address Compatibility (L) flag has been specified in the Binding Update, the home agent **MUST** do the same for the link-local address of the mobile node.

All fields in each such Neighbor Advertisement message **SHOULD** be set

in the same way they would be set by the mobile node itself if sending this Neighbor Advertisement [12] while at home, with the following exceptions:

- o The Target Address in the Neighbor Advertisement MUST be set to the specific IP address for the mobile node.
- o The Advertisement MUST include a Target Link-layer Address option specifying the home agent's link-layer address.
- o The Router (R) bit in the Advertisement MUST be set to zero.
- o The Solicited Flag (S) in the Advertisement MUST NOT be set, since it was not solicited by any Neighbor Solicitation.
- o The Override Flag (O) in the Advertisement MUST be set, indicating that the Advertisement SHOULD override any existing Neighbor Cache entry at any node receiving it.
- o The Source Address in the IPv6 header MUST be set to the home agent's IP address on the interface used to send the advertisement.

Any node on the home link receiving one of the Neighbor Advertisement messages described above will thus update its Neighbor Cache to associate the mobile node's address with the home agent's link layer address, causing it to transmit any future packets normally destined to the mobile node to the mobile node's home agent. Since multicasting on the local link (such as Ethernet) is typically not guaranteed to be reliable, the home agent MAY retransmit this Neighbor Advertisement message up to MAX_NEIGHBOR_ADVERTISEMENT (see [12]) times to increase its reliability. It is still possible that some nodes on the home link will not receive any of these Neighbor Advertisements, but these nodes will eventually be able to detect the link-layer address change for the mobile node's address, through use of Neighbor Unreachability Detection [12].

While a node is serving as a home agent for some mobile node, the home agent uses IPv6 Neighbor Discovery [12] to intercept unicast packets on the home link addressed to the mobile node. In order to intercept packets in this way, the home agent MUST act as a proxy for this mobile node, and reply to any received Neighbor Solicitations for it. When a home agent receives a Neighbor Solicitation, it MUST check if the Target Address specified in the message matches the address of any mobile node for which it has a Binding Cache entry marked as a home registration.

If such an entry exists in the home agent's Binding Cache, the home

agent **MUST** reply to the Neighbor Solicitation with a Neighbor Advertisement, giving the home agent's own link-layer address as the link-layer address for the specified Target Address. In addition, the Router (R) bit in the Advertisement **MUST** be set to zero. Acting as a proxy in this way allows other nodes on the mobile node's home link to resolve the mobile node's address, and allows the home agent to defend these addresses on the home link for Duplicate Address Detection [12].

10.4.2 Processing Intercepted Packets

For any packet sent to a mobile node from the mobile node's home agent (for which the home agent is the original sender of the packet), the home agent is operating as a correspondent node of the mobile node for this packet and the procedures described in Section 9.3.2 apply. The home agent then uses a routing header to route the packet to the mobile node by way of the primary care-of address in the home agent's Binding Cache.

While the mobile node is away from home, the home agent intercepts any packets on the home link addressed to the mobile node's home address, as described in Section 10.4.1. In order to forward each intercepted packet to the mobile node, the home agent **MUST** tunnel the packet to the mobile node using IPv6 encapsulation [15]. When a home agent encapsulates an intercepted packet for forwarding to the mobile node, the home agent sets the Source Address in the new tunnel IP header to the home agent's own IP address, and sets the Destination Address in the tunnel IP header to the mobile node's primary care-of address. When received by the mobile node, normal processing of the tunnel header [15] will result in decapsulation and processing of the original packet by the mobile node.

However, packets addressed to the mobile node's link-local address **MUST NOT** be tunneled to the mobile node. Instead, such a packet **MUST** be discarded, and the home agent **SHOULD** return an ICMP Destination Unreachable, Code 3, message to the packet's Source Address (unless this Source Address is a multicast address). Packets addressed to the mobile node's site-local address **SHOULD NOT** be tunneled to the mobile node by default.

Interception and tunneling of the following multicast addressed packets on the home network are only done if the home agent supports multicast group membership control messages from the mobile node as described in the next section. Tunneling of multicast packets to a mobile node follows similar limitations to those defined above for unicast packets addressed to the mobile node's link-local and site-local addresses. Multicast packets addressed to a multicast address with link-local scope [3], to which the mobile node is

subscribed, MUST NOT be tunneled to the mobile node; such packets SHOULD be silently discarded (after delivering to other local multicast recipients). Multicast packets addressed to a multicast address with scope larger than link-local but smaller than global (e.g., site-local and organization-local [3]), to which the mobile node is subscribed, SHOULD NOT be tunneled to the mobile node. Multicast packets addressed with a global scope to which the mobile node has successfully subscribed MUST be tunneled to the mobile node.

Before tunneling a packet to the mobile node, the home agent MUST perform any IPsec processing as indicated by the security policy data base.

10.4.3 Multicast Membership Control

This section is a prerequisite for the multicast data packet forwarding described in the previous section. If this support is not provided, multicast group membership control messages are silently ignored.

In order to forward multicast data packets from the home network to all the proper mobile nodes the home agent SHOULD be capable of receiving tunneled multicast group membership control information from the mobile node in order to determine which groups the mobile node has subscribed to. These multicast group membership messages are Listener Report messages specified MLD [17] or in other protocols such as [37].

The messages are issued by the mobile node but sent through the reverse tunnel to the home agent. These messages are issued whenever the mobile node decides to enable reception of packets for a multicast group or in response to an MLD Query from the home agent. The mobile node will also issue multicast group control messages to disable reception of multicast packets when it is no longer interested in receiving multicasts for a particular group.

To obtain the mobile node's current multicast group membership the home agent must periodically transmit MLD Query messages through the tunnel to the mobile node. These MLD periodic transmissions will ensure the home agent has an accurate record of the groups in which the mobile node is interested despite packet losses of the mobile node's MLD group membership messages.

All MLD packets are sent directly between the mobile node and the home agent. Since all these packets are destined to a link-scope multicast address and have a hop limit of 1, there is no direct forwarding of such packets between the home network and the mobile node. The MLD packets between the mobile node and the home agent are

encapsulated within the same tunnel header used for other packet flows between the mobile node and home agent.

Note that at this time, even though a link-local source is used on MLD packets, no functionality depends on these addresses being unique, nor do they elicit direct responses. All MLD messages are sent to multicast destinations. To avoid ambiguity on the home agent due to mobile nodes which may choose identical link-local source addresses for their MLD function it is necessary for the home agent to identify which mobile node was actually the issuer of a particular MLD message. This may be accomplished by noting which tunnel such an MLD arrived by, which IPsec SA was used, or by other distinguishing means.

This specification puts no requirement on how the functions in this section and the multicast forwarding in Section 10.4.2 are to be achieved. At the time of this writing it was thought that a full IPv6 multicast router function would be necessary on the home agent, but it may be possible to achieve the same effects through a "proxy MLD" application coupled with kernel multicast forwarding. This may be the subject of future specifications.

10.4.4 Stateful Address Autoconfiguration

This section describes how home agents support the use of stateful address autoconfiguration mechanisms such as DHCPv6 [29] from the mobile nodes. If this support is not provided, then the M and O bits must remain cleared on the Mobile Prefix Advertisement Messages. Any mobile node which sends DHCPv6 messages to the home agent without this support will not receive a response.

If DHCPv6 is used, packets are sent with link-local source addresses either to a link-scope multicast address or a link-local address. Mobile nodes desiring to locate a DHCPv6 service may reverse tunnel standard DHCPv6 packets to the home agent. Since these link-scope packets cannot be forwarded onto the home network it is necessary for the home agent to either implement a DHCPv6 relay agent or a DHCPv6 server function itself. The arriving tunnel or IPsec SA of DHCPv6 link-scope messages from the mobile node must be noted so that DHCPv6 responses may be sent back to the appropriate mobile node. DHCPv6 messages sent to the mobile node with a link-local destination must be tunneled within the same tunnel header used for other packet flows.

10.4.5 Handling Reverse Tunneled Packets

Unless a binding has been established between the mobile node and a correspondent node, traffic from the mobile node to the correspondent

node goes through a reverse tunnel. Home agents MUST support reverse tunneling as follows:

- o The tunneled traffic arrives to the home agent's address using IPv6 encapsulation [15].
- o Depending on the security policies used by the home agent, reverse tunneled packets MAY be discarded unless accompanied by a valid ESP header. The support for authenticated reverse tunneling allows the home agent to protect the home network and correspondent nodes from malicious nodes masquerading as a mobile node.
- o Otherwise, when a home agent decapsulates a tunneled packet from the mobile node, the home agent MUST verify that the Source Address in the tunnel IP header is the mobile node's primary care-of address. Otherwise any node in the Internet could send traffic through the home agent and escape ingress filtering limitations. This simple check forces the attacker to at least know the current location of the real mobile node and be able to defeat ingress filtering.

10.4.6 Protecting Return Routability Packets

The return routability procedure described in Section 5.2.5 assumes that the confidentiality of the Home Test Init and Home Test messages is protected as they are tunneled between the home agent to the mobile node. Therefore, the home agent MUST support tunnel mode IPsec ESP for the protection of packets belonging to the return routability procedure. Support for a non-null encryption transform and authentication algorithm MUST be available. It is not necessary to distinguish between different kinds of packets within the return routability procedure.

Security associations are needed to provide this protection. When the care-of address for the mobile node changes as a result of an accepted Binding Update, special treatment is needed for the next packets sent using these security associations. The home agent MUST set the new care-of address as the destination address of these packets, as if the outer header destination address in the security association had changed [21].

The above protection SHOULD be used with all mobile nodes. The use is controlled by configuration of the IPsec security policy database both at the mobile node and at the home agent.

As described earlier, the Binding Update and Binding Acknowledgement

messages require protection between the home agent and the mobile node. The Mobility Header protocol carries both these messages as well as the return routability messages. From the point of view of the security policy database these messages are indistinguishable. When IPsec is used to protect return routability signaling or payload packets, this protection **MUST** only be applied to the return routability packets entering the IPv6 encapsulated tunnel interface between the mobile node and the home agent. This can be achieved, for instance, by defining the security policy database entries specifically for the tunnel interface. That is, the policy entries are not generally applied on all traffic on the physical interface(s) of the nodes, but rather only on traffic that enters the tunnel. This makes use of per-interface security policy database entries [4], specific to the tunnel interface (the node's attachment to the tunnel [11]).

10.5 Dynamic Home Agent Address Discovery

This section describes how a home agent can help mobile nodes to discover the addresses of the home agents. The home agent keeps track of the other home agents on the same link, and responds to queries sent by the mobile node.

10.5.1 Receiving Router Advertisement Messages

For each link on which a router provides service as a home agent, the router maintains a Home Agents List recording information about all other home agents on that link. This list is used in the dynamic home agent address discovery mechanism, described in Section 10.5. The information for the list is learned through receipt of the periodic unsolicited multicast Router Advertisements, in a manner similar to the Default Router List conceptual data structure maintained by each host for Neighbor Discovery [12]. In the construction of the Home Agents List, the Router Advertisements are from each other home agent on the link, and the Home Agent (H) bit is set in them.

On receipt of a valid Router Advertisement, as defined in the processing algorithm specified for Neighbor Discovery [12], the home agent performs the following steps, in addition to any steps already required of it by Neighbor Discovery:

- o If the Home Agent (H) bit in the Router Advertisement is not set, delete the sending node's entry in the current Home Agents List (if one exists). Skip all the following steps.
- o Otherwise, extract the Source Address from the IP header of the Router Advertisement. This is the link-local IP address on this

link of the home agent sending this Advertisement [12].

- o Determine the preference for this home agent. If the Router Advertisement contains a Home Agent Information Option, then the preference is taken from the Home Agent Preference field in the option; otherwise, the default preference of 0 MUST be used.
- o Determine the lifetime for this home agent. If the Router Advertisement contains a Home Agent Information Option, then the lifetime is taken from the Home Agent Lifetime field in the option; otherwise, the lifetime specified by the Router Lifetime field in the Router Advertisement SHOULD be used.
- o If the link-local address of the home agent sending this Advertisement is already present in this home agent's Home Agents List and the received home agent lifetime value is zero, immediately delete this entry in the Home Agents List.
- o Otherwise, if the link-local address of the home agent sending this Advertisement is already present in the receiving home agent's Home Agents List, reset its lifetime and preference to the values determined above.
- o If the link-local address of the home agent sending this Advertisement is not already present in the Home Agents List maintained by the receiving home agent, and the lifetime for the sending home agent is non-zero, create a new entry in the list, and initialize its lifetime and preference to the values determined above.
- o If the Home Agents List entry for the link-local address of the home agent sending this Advertisement was not deleted as described above, determine any global address(es) of the home agent based on each Prefix Information option received in this Advertisement in which the Router Address (R) bit is set (Section 7.2). Add all such global addresses to the list of global addresses in this Home Agents List entry.

A home agent SHOULD maintain an entry in its Home Agents List for each valid home agent address until that entry's lifetime expires, after which time the entry MUST be deleted.

As described in Section 11.4.1, a mobile node attempts dynamic home agent address discovery by sending an ICMP Home Agent Address Discovery Request message to the Mobile IPv6 Home-Agents anycast address [16] for its home IP subnet prefix. A home agent receiving such a Home Agent Address Discovery Request message that is serving this subnet SHOULD return an ICMP Home Agent Address Discovery Reply

message to the mobile node, with the Source Address of the Reply packet set to one of the global unicast addresses of the home agent. The Home Agent Addresses field in the Reply message is constructed as follows:

- o The Home Agent Addresses field SHOULD contain all global IP addresses for each home agent currently listed in this home agent's own Home Agents List (Section 10.1).
- o The IP addresses in the Home Agent Addresses field SHOULD be listed in order of decreasing preference values, based either on the respective advertised preference from a Home Agent Information option or on the default preference of 0 if no preference is advertised (or on the configured home agent preference for this home agent itself).
- o Among home agents with equal preference, their IP addresses in the Home Agent Addresses field SHOULD be listed in an order randomized with respect to other home agents with equal preference, each time a Home Agent Address Discovery Reply message is returned by this home agent.
- o If more than one global IP address is associated with a home agent, these addresses SHOULD be listed in a randomized order.
- o The home agent SHOULD reduce the number of home agent IP addresses so that the packet fits within the minimum IPv6 MTU [11]. The home agent addresses selected for inclusion in the packet SHOULD be those from the complete list with the highest preference. This limitation avoids the danger of the Reply message packet being fragmented (or rejected by an intermediate router with an ICMP Packet Too Big message [14]).

10.6 Sending Prefix Information to the Mobile Node

10.6.1 List of Home Network Prefixes

Mobile IPv6 arranges to propagate relevant prefix information to the mobile node when it is away from home, so that it may be used in mobile node home address configuration, and in network renumbering. In this mechanism, mobile nodes away from home receive Mobile Prefix Advertisements messages. These messages include Prefix Information Options for the prefixes configured on the home subnet interface(s) of the home agent.

If there are multiple home agents, differences in the advertisements sent by different home agents can lead to an inability to use a

particular home address when changing to another home agent. In order to ensure that the mobile nodes get the same information from different home agents, it is desired that all the home agents on the same link be configured in the same manner.

To support this, the home agent monitors prefixes advertised by itself and other home agents on the home link. In RFC 2461 [12] it is acceptable for two routers to advertise different sets of prefixes on the same link. For home agents such differences should be detected since for a given home address the mobile node communicates only with one home agent at a time and the mobile node needs to know the full set of prefixes assigned to the home link. All other comparisons of Router Advertisements are as specified in Section 6.2.7 of RFC 2461.

10.6.2 Scheduling Prefix Deliveries

A home agent serving a mobile node will schedule the delivery of new prefix information to that mobile node when any of the following conditions occur:

MUST:

- o The state of the flags changes for the prefix of the mobile node's registered home address.
- o The valid or preferred lifetime is reconfigured or changes for any reason other than advancing real time.
- o The mobile node requests the information with a Mobile Prefix Solicitation (see Section 11.4.2).

SHOULD:

- o A new prefix is added to the home subnet interface(s) of the home agent.

MAY:

- o The valid or preferred lifetime or the state of the flags changes for a prefix which is not used in any Binding Cache entry for this mobile node.

The home agent uses the following algorithm to determine when to send prefix information to the mobile node.

- o If a mobile node sends a solicitation, answer right away.

- o If no Mobile Prefix Advertisement has been sent to the mobile node in the last MaxMobPfxAdvInterval (see Section 13) seconds, then ensure that a transmission is scheduled. The actual transmission time is randomized as described below.
- o If a prefix matching the mobile node's home registration is added on the home subnet interface, or if its information changes in any way that does not deprecate the mobile node's address, ensure that a transmission is scheduled. The actual transmission time is randomized as described below.
- o If a home registration expires, cancel any scheduled advertisements to the mobile node.

The list of prefixes is sent in its entirety in all cases.

If the home agent already has scheduled the transmission of a Mobile Prefix Advertisement to the mobile node, the home agent replaces the advertisement with a new one, to be sent at the scheduled time.

Otherwise, the home agent computes a fresh value for RAND_ADV_DELAY, the offset from the current time for the scheduled transmission as follows. First calculate the maximum delay for the scheduled Advertisement:

$$\text{MaxScheduleDelay} = \min (\text{MaxMobPfxAdvInterval}, \text{Preferred Lifetime}),$$

where MaxMobPfxAdvInterval is as defined in Section 12. Then compute the final delay for the advertisement:

$$\text{RAND_ADV_DELAY} = \text{MinMobPfxAdvInterval} + \\ (\text{rand}() \% \text{abs}(\text{MaxScheduleDelay} - \text{MinMobPfxAdvInterval}))$$

Here rand() returns a random integer value in the range of 0 to the maximum possible integer value. This computation is expected to alleviate bursts of advertisements when prefix information changes. In addition, a home agent MAY further reduce the rate of packet transmission by further delaying individual advertisements, if needed to avoid overwhelming local network resources. The home agent SHOULD periodically continue to retransmit an unsolicited Advertisement to the mobile node, until it is acknowledged by the receipt of a Mobile Prefix Solicitation from the mobile node.

The home agent MUST wait PREFIX_ADV_TIMEOUT (see Section 12) before

the first retransmission, and double the retransmission wait time for every succeeding retransmission, up until a maximum of PREFIX_ADV_RETRIES attempts (see Section 12). If the mobile node's bindings expire before the matching Binding Update has been received, then the home agent MUST NOT attempt any more retransmissions, even if not all PREFIX_ADV_RETRIES have been retransmitted. If the mobile node sends another Binding Update without returning home in the meantime, the home agent SHOULD again begin transmitting the unsolicited Advertisement.

If some condition as described above occurs on the home link and causes another Prefix Advertisement to be sent to the mobile node, before the mobile node acknowledges a previous transmission, the home agent SHOULD combine any Prefix Information options in the unacknowledged Mobile Prefix Advertisement into a new Advertisement. The home agent discards the old Advertisement.

10.6.3 Sending Advertisements

When sending a Mobile Prefix Advertisement to the mobile node, the home agent MUST construct the packet as follows:

- o The Source Address in the packet's IPv6 header MUST be set to the home agent's IP address to which the mobile node addressed its current home registration, or its default global home agent address if no binding exists.
- o If the advertisement was solicited, it MUST be destined to the source address of the solicitation. If it was triggered by prefix changes or renumbering, the advertisement's destination will be the mobile node's home address in the binding which triggered the rule.
- o A type 2 routing header MUST be included with the mobile node's home address.
- o IPsec headers MUST be supported and SHOULD be used.
- o The home agent MUST send the packet as it would any other unicast IPv6 packet that it originates.
- o Set the Managed Address Configuration (M) flag if the corresponding flag has been set in any of the Router Advertisements from which the prefix information has been learned (including the ones sent by this home agent).
- o Set the Other Stateful Configuration (O) flag if the corresponding flag has been set in any of the Router Advertisements from which

the prefix information has been learned (including the ones sent by this home agent).

10.6.4 Lifetimes for Changed Prefixes

As described in Section 10.3.1, the lifetime returned by the home agent in a Binding Acknowledgement **MUST** be no greater than the remaining valid lifetime for the subnet prefix in the mobile node's home address. This limit on the binding lifetime serves to prohibit use of a mobile node's home address after it becomes invalid.

11. Mobile Node Operation

11.1 Conceptual Data Structures

Each mobile node **MUST** maintain a Binding Update List.

The Binding Update List records information for each Binding Update sent by this mobile node, for which the lifetime of the binding has not yet expired. The Binding Update List includes all bindings sent by the mobile node either to its home agent or correspondent nodes. It also contains Binding Updates which are waiting for the completion of the return routability procedure before they can be sent. However, for multiple Binding Updates sent to the same destination address, the Binding Update List contains only the most recent Binding Update (i.e., with the greatest Sequence Number value) sent to that destination. The Binding Update List **MAY** be implemented in any manner consistent with the external behavior described in this document.

Each Binding Update List entry conceptually contains the following fields:

- o The IP address of the node to which a Binding Update was sent.
- o The home address for which that Binding Update was sent.
- o The care-of address sent in that Binding Update. This value is necessary for the mobile node to determine if it has sent a Binding Update giving its new care-of address to this destination after changing its care-of address.
- o The initial value of the Lifetime field sent in that Binding Update.
- o The remaining lifetime of that binding. This lifetime is initialized from the Lifetime value sent in the Binding Update and is decremented until it reaches zero, at which time this entry **MUST** be deleted from the Binding Update List.
- o The maximum value of the Sequence Number field sent in previous Binding Updates to this destination. The Sequence Number field is 16 bits long, and all comparisons between Sequence Number values **MUST** be performed modulo 2^{16} (see Section 9.5.1).
- o The time at which a Binding Update was last sent to this destination, as needed to implement the rate limiting restriction for sending Binding Updates.

- o The state of any retransmissions needed for this Binding Update. This state includes the time remaining until the next retransmission attempt for the Binding Update, and the current state of the exponential back-off mechanism for retransmissions.
- o A flag specifying whether or not future Binding Updates should be sent to this destination. The mobile node sets this flag in the Binding Update List entry when it receives an ICMP Parameter Problem, Code 1, error message in response to a return routability message or Binding Update sent to that destination, as described in Section 11.3.5.

The Binding Update List is used to determine whether a particular packet is sent directly to the correspondent node or tunneled via the home agent (see Section 11.3.1).

The Binding Update list also conceptually contains the following data related to running the return routability procedure. This data is relevant only for Binding Updates sent to correspondent nodes.

- o The time at which a Home Test Init or Care-of Test Init message was last sent to this destination, as needed to implement the rate limiting restriction for the return routability procedure.
- o The state of any retransmissions needed for this return routability procedure. This state includes the time remaining until the next retransmission attempt and the current state of the exponential back-off mechanism for retransmissions.
- o Cookie values used in the Home Test Init and Care-of Test Init messages.
- o Home and care-of keygen tokens received from the correspondent node.
- o Home and care-of nonce indices received from the correspondent node.
- o The time at which each of the tokens and nonces was received from this correspondent node, as needed to implement reuse while moving.

11.2 Processing Mobility Headers

All IPv6 mobile nodes **MUST** observe the rules described in Section 9.2 when processing Mobility Headers.

11.3 Packet Processing

11.3.1 Sending Packets While Away from Home

While a mobile node is away from home, it continues to use its home address, as well as also using one or more care-of addresses. When sending a packet while away from home, a mobile node MAY choose among these in selecting the address that it will use as the source of the packet, as follows:

- o Protocols layered over IP will generally treat the mobile node's home address as its IP address for most packets. For packets sent that are part of transport-level connections established while the mobile node was at home, the mobile node MUST use its home address. Likewise, for packets sent that are part of transport-level connections that the mobile node may still be using after moving to a new location, the mobile node SHOULD use its home address in this way. If a binding exists, the mobile node SHOULD send the packets directly to the correspondent node. Otherwise, if a binding does not exist, the mobile node MUST use reverse tunneling.
- o The mobile node MAY choose to directly use one of its care-of addresses as the source of the packet, not requiring the use of a Home Address option in the packet. This is particularly useful for short-term communication that may easily be retried if it fails. Using the mobile node's care-of address as the source for such queries will generally have a lower overhead than using the mobile node's home address, since no extra options need be used in either the query or its reply. Such packets can be routed normally, directly between their source and destination without relying on Mobile IPv6. If application running on the mobile node has no particular knowledge that the communication being sent fits within this general type of communication, however, the mobile node should not use its care-of address as the source of the packet in this way.

The choice of the most efficient communications method is application specific, and outside the scope of this specification. The APIs necessary for controlling the choice are also out of scope.

- o While not at its home link, the mobile node MUST NOT use the Home Address destination option when communicating with link-local or site-local peers, if the scope of the home address is larger than the scope of the peer's address.

Similarly, the mobile node MUST NOT use the Home Address

destination option for IPv6 Neighbor Discovery [12] packets.

Detailed operation of these cases is described later in this section and also discussed in [31].

For packets sent by a mobile node while it is at home, no special Mobile IPv6 processing is required. Likewise, if the mobile node uses any address other than any of its home addresses as the source of a packet sent while away from home no special Mobile IPv6 processing is required. In either case, the packet is simply addressed and transmitted in the same way as any normal IPv6 packet.

For packets sent by the mobile node sent while away from home using the mobile node's home address as the source, special Mobile IPv6 processing of the packet is required. This can be done in the following two ways:

Route Optimization

This manner of delivering packets does not require going through the home network, and typically will enable faster and more reliable transmission.

The mobile node needs to ensure that there exists a Binding Cache entry for its home address so that the correspondent node can process the packet (Section 9.3.1 specifies the rules for Home Address Destination Option Processing at a correspondent node). The mobile node SHOULD examine its Binding Update List for an entry which fulfills the following conditions:

- * The Source Address field of the packet being sent is equal to the home address in the entry.
- * The Destination Address field of the packet being sent is equal to the address of the correspondent node in the entry.
- * One of the current care-of addresses of the mobile node appears as the care-of address in the entry.
- * The entry indicates that a binding has been successfully created.
- * The remaining lifetime of the binding is greater than zero.

When these conditions are met, the mobile node knows that the correspondent node has a suitable Binding Cache entry.

A mobile node SHOULD arrange to supply the home address in a Home Address option, and MUST set the IPv6 header's Source Address field to the care-of address which the mobile node has registered to be used with this correspondent node. The correspondent node will then use the address supplied in the Home Address option to serve the function traditionally done by the Source IP address in the IPv6 header. The mobile node's home address is then supplied to higher protocol layers and applications.

Specifically:

- * Construct the packet using the mobile node's home address as the packet's Source Address, in the same way as if the mobile node were at home. This includes the calculation of upper layer checksums using the home address as the value of the source.
- * Insert a Home Address option into the packet, with the Home Address field copied from the original value of the Source Address field in the packet.
- * Change the Source Address field in the packet's IPv6 header to one of the mobile node's care-of addresses. This will typically be the mobile node's current primary care-of address, but MUST be an address assigned to the interface on the link being used.

By using the care-of address as the Source Address in the IPv6 header, with the mobile node's home address instead in the Home Address option, the packet will be able to safely pass through any router implementing ingress filtering [26].

Reverse Tunneling

This is the mechanism which tunnels the packets via the home agent. It is not as efficient as the above mechanism, but is needed if there is no binding yet with the correspondent node.

This mechanism is used for packets that have the mobile node's home address as the Source Address in the IPv6 header, or with multicast control protocol packets as described in Section 11.3.4. Specifically:

- * The packet is sent to the home agent using IPv6 encapsulation [15].

- * The Source Address in the tunnel packet is the primary care-of address as registered with the home agent.
- * The Destination Address in the tunnel packet is the home agent's address.

Then, the home agent will pass the encapsulated packet to the correspondent node.

11.3.2 Interaction with Outbound IPsec Processing

This section sketches the interaction between outbound Mobile IPv6 processing and outbound IP Security (IPsec) processing for packets sent by a mobile node while away from home. Any specific implementation MAY use algorithms and data structures other than those suggested here, but its processing MUST be consistent with the effect of the operation described here and with the relevant IPsec specifications. In the steps described below, it is assumed that IPsec is being used in transport mode [4] and that the mobile node is using its home address as the source for the packet (from the point of view of higher protocol layers or applications, as described in Section 11.3.1):

- o The packet is created by higher layer protocols and applications (e.g., by TCP) as if the mobile node were at home and Mobile IPv6 were not being used.
- o Determine the outgoing interface for the packet. (Note that the selection between reverse tunneling and route optimization may imply different interfaces, particularly if tunnels are considered interfaces as well.)
- o As part of outbound packet processing in IP, the packet is compared against the IPsec security policy database to determine what processing is required for the packet [4].
- o If IPsec processing is required, the packet is either mapped to an existing Security Association (or SA bundle), or a new SA (or SA bundle) is created for the packet, according to the procedures defined for IPsec.
- o Since the mobile node is away from home, the mobile is either using reverse tunneling or route optimization to reach the correspondent node.

If reverse tunneling is used, the packet is constructed in the

normal manner and then tunneled through the home agent.

If route optimization is in use, the mobile node inserts a Home Address destination option into the packet, replacing the Source Address in the packet's IP header with the care-of address used with this correspondent node, as described in Section 11.3.1. The Destination Options header in which the Home Address destination option is inserted **MUST** appear in the packet after the routing header, if present, and before the IPsec (AH [5] or ESP [6]) header, so that the Home Address destination option is processed by the destination node before the IPsec header is processed.

Finally, once the packet is fully assembled, the necessary IPsec authentication (and encryption, if required) processing is performed on the packet, initializing the Authentication Data in the IPsec header.

RFC 2402 treatment of destination options is extended as follows. The AH authentication data **MUST** be calculated as if the following were true:

- * the IPv6 source address in the IPv6 header contains the mobile node's home address,
 - * the Home Address field of the Home Address destination option (Section 6.3) contains the new care-of address.
- o This allows, but does not require, the receiver of the packet containing a Home Address destination option to exchange the two fields of the incoming packet to reach the above situation, simplifying processing for all subsequent packet headers. However, such an exchange is not required, as long as the result of the authentication calculation remains the same.

When an automated key management protocol is used to create new security associations for a peer, it is important to ensure that the peer can send the key management protocol packets to the mobile node. This may not be possible if the peer is the home agent of the mobile node, and the purpose of the security associations would be to send a Binding Update to the home agent. Packets addressed to the home address of the mobile node cannot be used before the Binding Update has been processed. For the default case of using IKE [9] as the automated key management protocol, such problems can be avoided by the following requirements when communicating with its home agent:

- o When the mobile node is away from home, it **MUST** use its care-of address as the Source Address of all packets it sends as part of the key management protocol (without use of Mobile IPv6 for these

packets, as suggested in Section 11.3.1).

- o In addition, for all security associations bound to the mobile node's home address established by IKE, the mobile node **MUST** include an ISAKMP Identification Payload [8] in the IKE exchange, giving the mobile node's home address as the initiator of the Security Association [7].

The Key Management Mobility Capability (K) bit in Binding Updates and Acknowledgements can be used avoid the need to rerun IKE upon movements.

11.3.3 Receiving Packets While Away from Home

While away from home, a mobile node will receive packets addressed to its home address, by one of two methods:

- o Packets sent by a correspondent node that does not have a Binding Cache entry for the mobile node, will be sent to the home address, captured by the home agent and tunneled to the mobile node
- o Packets sent by a correspondent node that has a Binding Cache entry for the mobile node that contains the mobile node's current care-of address, will be sent by the correspondent node using a type 2 routing header. The packet will be addressed to the mobile node's care-of address, with the final hop in the routing header directing the packet to the mobile node's home address; the processing of this last hop of the routing header is entirely internal to the mobile node, since the care-of address and home address are both addresses within the mobile node.

For packets received by the first method, the mobile node **MUST** check that the IPv6 source address of the tunneled packet is the IP address of its home agent. In this method the mobile node may also send a Binding Update to the original sender of the packet, as described in Section 11.7.2, subject to the rate limiting defined in Section 11.8. The mobile node **MUST** also process the received packet in the manner defined for IPv6 encapsulation [15], which will result in the encapsulated (inner) packet being processed normally by upper-layer protocols within the mobile node, as if it had been addressed (only) to the mobile node's home address.

For packets received by the second method, the following rules will result in the packet being processed normally by upper-layer protocols within the mobile node, as if it had been addressed to the mobile node's home address.

A node receiving a packet addressed to itself (i.e., one of the

node's addresses is in the IPv6 destination field) follows the next header chain of headers and processes them. When it encounters a type 2 routing header during this processing it performs the following checks. If any of these checks fail the node **MUST** silently discard the packet.

- o The length field in the routing header is exactly 2.
- o The segments left field in the routing header is 1 on the wire. (But implementations may process the routing header so that the value may become 0 after the routing header has been processed, but before the rest of the packet is processed.)
- o The Home Address field in the routing header is one of the node's home addresses, if the segments left field was 1. Thus, in particular the address field is required to be a unicast routable address.

Once the above checks have been performed, the node swaps the IPv6 destination field with the Home Address field in the routing header, decrements segments left by one from the value it had on the wire, and resubmits the packet to IP for processing the next header. Conceptually this follows the same model as in RFC 2460. However, in the case of type 2 routing header this can be simplified since it is known that the packet will not be forwarded to a different node.

The definition of AH requires the sender to calculate the AH integrity check value of a routing header in a way as it appears in the receiver after it has processed the header. Since IPsec headers follow the routing header, any IPsec processing will operate on the packet with the home address in the IP destination field and segments left being zero. Thus, the AH calculations at the sender and receiver will have an identical view of the packet.

11.3.4 Routing Multicast Packets

A mobile node that is connected to its home link functions in the same way as any other (stationary) node. Thus, when it is at home, a mobile node functions identically to other multicast senders and receivers. This section therefore describes the behavior of a mobile node that is not on its home link.

In order to receive packets sent to some multicast group, a mobile node must join that multicast group. One method by which a mobile node **MAY** join the group is via a (local) multicast router on the foreign link being visited. In this case, the mobile node **MUST** use its care-of address and **MUST NOT** use the Home Address destination option when sending MLD packets [17].

Alternatively, a mobile node MAY join multicast groups via a bi-directional tunnel to its home agent. The mobile node tunnels its multicast group membership control packets (such as those defined in [17] or in [37]) to its home agent, and the home agent forwards multicast packets down the tunnel to the mobile node. A mobile node MUST NOT tunnel multicast group membership control packets until (1) the mobile node has a binding in place at the home agent, and (2) the latter sends at least one such multicast group membership control packet via the tunnel. Once this condition is true, the mobile node SHOULD assume it does not change as long as the binding does not expire.

A mobile node that wishes to send packets to a multicast group also has two options:

1. Send directly on the foreign link being visited.

The application is aware of the care-of address and uses it as a source address for multicast traffic, just like it would use a stationary address. The mobile node MUST NOT use Home Address destination option in such traffic.

2. Send via a tunnel to its home agent.

Because multicast routing in general depends upon the Source Address used in the IPv6 header of the multicast packet, a mobile node that tunnels a multicast packet to its home agent MUST use its home address as the IPv6 Source Address of the inner multicast packet.

Note that direct sending from the foreign link is only applicable while the mobile node is at that foreign link. This is because the associated multicast tree is specific to that source location and any change of location and source address will invalidate the source specific tree or branch and the application context of the other multicast group members.

This specification does not provide mechanisms to enable such local multicast session to survive hand-off, and to seamlessly continue from a new care-of address on each new foreign link. Any such mechanism, developed as an extension to this specification, needs to take into account the impact of fast moving mobile nodes on the Internet multicast routing protocols and their ability to maintain the integrity of source specific multicast trees and branches.

While the use of bidirectional tunneling can ensure that multicast trees are independent of the mobile nodes movement, in some case such tunneling can have adverse affects. The latency of specific types of

multicast applications such as multicast based discovery protocols will be affected when the round-trip time between the foreign subnet and the home agent is significant compared to that of the topology to be discovered. In addition, the delivery tree from the home agent in such circumstances relies on unicast encapsulation from the agent to the mobile node and is therefore bandwidth inefficient compared to the native multicast forwarding in the foreign multicast system.

11.3.5 Receiving ICMP Error Messages

Any node that does not recognize the Mobility header will return an ICMP Parameter Problem, Code 1, message to the sender of the packet. If the mobile node receives such an ICMP error message in response to a return routability procedure or Binding Update, it SHOULD record in its Binding Update List that future Binding Updates SHOULD NOT be sent to this destination. Such Binding Update List entries SHOULD be removed after a period of time, in order to allow for retrying route optimization.

New Binding Update List entries MUST NOT be created as a result of receiving ICMP error messages.

Correspondent nodes that have participated in the return routability procedure MUST implement the ability to correctly process received packets containing a Home Address destination option. Therefore, correctly implemented correspondent nodes should always be able to recognize Home Address options. If a mobile node receives an ICMP Parameter Problem, Code 2, message from some node indicating that it does not support the Home Address option, the mobile node SHOULD log the error and then discard the ICMP message.

11.3.6 Receiving Binding Error Messages

When a mobile node receives a packet containing a Binding Error message, it should first check if the mobile node has a Binding Update List entry for the source of the Binding Error message. If the mobile node does not have such an entry, it MUST ignore the message. This is necessary to prevent a waste of resources on e.g. return routability procedure due to spoofed Binding Error messages.

Otherwise, if the message Status field was 1 (unknown binding for Home Address destination option), the mobile node should perform one of the following two actions:

- o If the mobile node has recent upper layer progress information that indicates communications with the correspondent node are progressing, it MAY ignore the message. This can be done in order to limit the damage that spoofed Binding Error messages can cause

to ongoing communications.

- o If the mobile node has no upper layer progress information, it **MUST** remove the entry and route further communications through the home agent. It **MAY** also optionally start a return routability procedure (see Section 5.2).

If the message Status field was 2 (unrecognized MH Type value), the mobile node should perform one of the following two actions:

- o If the mobile node is not expecting an acknowledgement or response from the correspondent node, the mobile node **SHOULD** ignore this message.
- o Otherwise, the mobile node **SHOULD** cease the use of any extensions to this specification. If no extensions had been used, the mobile node should cease the attempt to use route optimization.

11.4 Home Agent and Prefix Management

11.4.1 Dynamic Home Agent Address Discovery

Sometimes, when the mobile node needs to send a Binding Update to its home agent to register its new primary care-of address, as described in Section 11.7.1, the mobile node may not know the address of any router on its home link that can serve as a home agent for it. For example, some nodes on its home link may have been reconfigured while the mobile node has been away from home, such that the router that was operating as the mobile node's home agent has been replaced by a different router serving this role.

In this case, the mobile node **MAY** attempt to discover the address of a suitable home agent on its home link. To do so, the mobile node sends an ICMP Home Agent Address Discovery Request message to the Mobile IPv6 Home-Agents anycast address [16] for its home subnet prefix. As described in Section 10.5, the home agent on its home link that receives this Request message will return an ICMP Home Agent Address Discovery Reply message. This message gives the addresses for the home agents operating on the home link.

The mobile node, upon receiving this Home Agent Address Discovery Reply message, **MAY** then send its home registration Binding Update to any of the unicast IP addresses listed in the Home Agent Addresses field in the Reply. For example, the mobile node **MAY** attempt its home registration to each of these addresses, in turn, until its registration is accepted. The mobile node sends a Binding Update to an address and waits for the matching Binding Acknowledgement, moving

on to the next address if there is no response. The mobile node **MUST**, however, wait at least `InitialBindackTimeoutFirstReg` seconds (see Section 13) before sending a Binding Update to the next home agent. In trying each of the returned home agent addresses, the mobile node **SHOULD** try each in the order listed in the Home Agent Addresses field in the received Home Agent Address Discovery Reply message.

If the mobile node has a current registration with some home agent (the Lifetime for that registration has not yet expired), then the mobile node **MUST** attempt any new registration first with that home agent. If that registration attempt fails (e.g., times out or is rejected), the mobile node **SHOULD** then reattempt this registration with another home agent. If the mobile node knows of no other suitable home agent, then it **MAY** attempt the dynamic home agent address discovery mechanism described above.

If, after a mobile node transmits a Home Agent Address Discovery Request message to the Home Agents Anycast address, it does not receive a corresponding Home Agent Address Discovery Reply message within `INITIAL_DHAAD_TIMEOUT` (see Section 12) seconds, the mobile node **MAY** retransmit the same Request message to the same anycast address. This retransmission **MAY** be repeated up to a maximum of `DHAAD_RETRIES` (see Section 12) attempts. Each retransmission **MUST** be delayed by twice the time interval of the previous retransmission.

11.4.2 Sending Mobile Prefix Solicitations

When a mobile node has a home address that is about to become invalid, it **SHOULD** send a Mobile Prefix Solicitation to its home agent in an attempt to acquire fresh routing prefix information. The new information also enables the mobile node to participate in renumbering operations affecting the home network, as described in Section 10.6.

The mobile node **MUST** use the Home Address destination option to carry its home address. The mobile node **MUST** support and **SHOULD** use IPsec to protect the solicitation. The mobile node **MUST** set the Identifier field in the ICMP header to a random value.

As described in Section 11.7.2, Binding Updates sent by the mobile node to other nodes **MUST** use a lifetime no greater than the remaining lifetime of its home registration of its primary care-of address. The mobile node **SHOULD** further limit the lifetimes that it sends on any Binding Updates to be within the remaining valid lifetime (see Section 10.6.2) for the prefix in its home address.

When the lifetime for a changed prefix decreases, and the change

would cause cached bindings at correspondent nodes in the Binding Update List to be stored past the newly shortened lifetime, the mobile node **MUST** issue a Binding Update to all such correspondent nodes.

These limits on the binding lifetime serve to prohibit use of a mobile node's home address after it becomes invalid.

11.4.3 Receiving Mobile Prefix Advertisements

Section 10.6 describes the operation of a home agent to support boot time configuration and renumbering a mobile node's home subnet while the mobile node is away from home. The home agent sends Mobile Prefix Advertisements to the mobile node while away from home, giving "important" Prefix Information options that describe changes in the prefixes in use on the mobile node's home link.

The Mobile Prefix Solicitation is similar to the Router Solicitation used in Neighbor Discovery [12], except it is routed from the mobile node on the visited network to the home agent on the home network by usual unicast routing rules.

When a mobile node receives a Mobile Prefix Advertisement, it **MUST** validate it according to the following test:

- o The Source Address of the IP packet carrying the Mobile Prefix Advertisement is the same as the home agent address to which the mobile node last sent an accepted home registration Binding Update to register its primary care-of address. Otherwise, if no such registrations have been made, it **SHOULD** be the mobile node's stored home agent address, if one exists. Otherwise, if the mobile node has not yet discovered its home agent's address, it **MUST NOT** accept Mobile Prefix Advertisements.
- o The packet **MUST** have a type 2 routing header and **SHOULD** be protected by an IPsec header as described in Section 5.4 and Section 6.8.
- o If the ICMP Identifier value matches the ICMP Identifier value of the most recently sent Mobile Prefix Solicitation and no other advertisement has yet been received for this value, then the advertisement is considered to be solicited and will be processed further.

Otherwise, the advertisement is unsolicited, and **MUST** be silently discarded. In this case the mobile node **SHOULD** send a Mobile Prefix Solicitation.

Any received Mobile Prefix Advertisement not meeting these tests **MUST** be silently discarded.

For an accepted Mobile Prefix Advertisement, the mobile node **MUST** process Managed Address Configuration (M), Other Stateful Configuration (O), and the Prefix Information Options as if they arrived in a Router Advertisement [12] on the mobile node's home link. (This specification does not, however, describe how to acquire home addresses through stateful protocols.) Such processing may result in the mobile node configuring a new home address, although due to separation between preferred lifetime and valid lifetime, such changes should not affect most communications by the mobile node, in the same way as for nodes that are at home.

This specification assumes that any security associations and security policy entries that may be needed for new prefixes have been pre-configured in the mobile node. Note that while dynamic key management avoids the need to create new security associations, it is still necessary to add policy entries to protect the communications involving the home address(es). Mechanisms for automatic set-up of these entries are outside the scope of this specification.

11.5 Movement

11.5.1 Movement Detection

The primary goal of movement detection is to detect L3 handovers. This section does not attempt to specify a fast movement detection algorithm which will function optimally for all types of applications, link-layers and deployment scenarios; instead, it describes a generic method that uses the facilities of IPv6 Neighbor Discovery, including Router Discovery and Neighbor Unreachability Detection. At the time of this writing, this method is considered well enough understood to recommend for standardization, however it is expected that future versions of this specification or other specifications may contain updated versions of the movement detection algorithm that have better performance.

Generic movement detection uses Neighbor Unreachability Detection to detect when the default router is no longer bi-directionally reachable, in which case the mobile node must discover a new default router (usually on a new link). However, this detection only occurs when the mobile node has packets to send, and in the absence of frequent Router Advertisements or indications from the link-layer, the mobile node might become unaware of an L3 handover that occurred. Therefore, the mobile node should supplement this method with other information whenever it is available to the mobile node (e.g., from lower protocol layers).

When the mobile node detects an L3 handover, it performs Duplicate Address Detection [13] on its link-local address, selects a new default router as a consequence of Router Discovery, and then performs Prefix Discovery with that new router to form new care-of address(es) as described in Section 11.5.2. It then registers its new primary care-of address with its home agent as described in Section 11.7.1. After updating its home registration, the mobile node then updates associated mobility bindings in correspondent nodes that it is performing route optimization with as specified in Section 11.7.2.

Due to the temporary packet flow disruption and signaling overhead involved in updating mobility bindings, the mobile node should avoid performing an L3 handover until it is strictly necessary. Specifically, when the mobile node receives a Router Advertisement from a new router that contains a different set of on-link prefixes, if the mobile node detects that the currently selected default router on the old link is still bi-directionally reachable, it should generally continue to use the old router on the old link rather than switch away from it to use a new default router.

Mobile nodes can use the information in received Router Advertisements to detect L3 handovers. In doing so the mobile node needs to consider the following issues:

- o There might be multiple routers on the same link, thus hearing a new router does not necessarily constitute an L3 handover.
- o When there are multiple routers on the same link they might advertise different prefixes. Thus even hearing a new router with a new prefix might not be a reliable indication of an L3 handover.
- o The link-local addresses of routers are not globally unique, hence after completing an L3 handover the mobile node might continue to receive Router Advertisements with the same link-local source address. This might be common if routers use the same link-local address on multiple interfaces. This issue can be avoided when routers use the Router Address (R) bit, since that provides a global address of the router.

In addition, the mobile node should consider the following events as indications that an L3 handover may have occurred. Upon receiving such indications, the mobile node needs to perform Router Discovery to discover routers and prefixes on the new link, as described in Section 6.3.7 of RFC 2461 [12].

- o If Router Advertisements that the mobile node receives include an Advertisement Interval option, the mobile node may use its

Advertisement Interval field as an indication of the frequency with which it should expect to continue to receive future Advertisements from that router. This field specifies the minimum rate (the maximum amount of time between successive Advertisements) that the mobile node should expect. If this amount of time elapses without the mobile node receiving any Advertisement from this router, the mobile node can be sure that at least one Advertisement sent by the router has been lost. The mobile node can then implement its own policy to determine how many lost Advertisements from its current default router constitute an L3 handover indication.

- o Neighbor Unreachability Detection determines that the default router is no longer reachable.
- o With some types of networks, notification that an L2 handover has occurred might be obtained from lower layer protocols or device driver software within the mobile node. While further details around handling L2 indications as movement hints is an item for further study, at the time of writing this specification the following is considered reasonable:

An L2 handover indication may or may not imply L2 movement and L2 movement may or may not imply L3 movement; the correlations might be a function of the type of L2 but might also be a function of actual deployment of the wireless topology.

Unless it is well-known that an L2 handover indication is likely to imply L3 movement, instead of immediately multicasting a router solicitation it may be better to attempt to verify whether the default router is still bi-directionally reachable. This can be accomplished by sending a unicast Neighbor Solicitation and waiting for a Neighbor Advertisement with the solicited flag set. Note that this is similar to Neighbor Unreachability detection but it does not have the same state machine, such as the STALE state.

If the default router does not respond to the Neighbor Solicitation it makes sense to proceed to multicasting a Router Solicitation.

11.5.2 Forming New Care-of Addresses

After detecting that it has moved a mobile node SHOULD generate a new primary care-of address using normal IPv6 mechanisms. This SHOULD also be done when the current primary care-of address becomes deprecated. A mobile node MAY form a new primary care-of address at any time, but a mobile node MUST NOT send a Binding Update about a

new care-of address to its home agent more than MAX_UPDATE_RATE times within a second.

In addition, a mobile node MAY form new non-primary care-of addresses even when it has not switched to a new default router. A mobile node can have only one primary care-of address at a time (which is registered with its home agent), but it MAY have an additional care-of address for any or all of the prefixes on its current link. Furthermore, since a wireless network interface may actually allow a mobile node to be reachable on more than one link at a time (i.e., within wireless transmitter range of routers on more than one separate link), a mobile node MAY have care-of addresses on more than one link at a time. The use of more than one care-of address at a time is described in Section 11.5.3.

As described in Section 4, in order to form a new care-of address, a mobile node MAY use either stateless [13] or stateful (e.g., DHCPv6 [29]) Address Autoconfiguration. If a mobile node needs to use a source address (other than the unspecified address) in packets sent as a part of address autoconfiguration, it MUST use an IPv6 link-local address rather than its own IPv6 home address.

RFC 2462 [13] specifies that in normal processing for Duplicate Address Detection, the node SHOULD delay sending the initial Neighbor Solicitation message by a random delay between 0 and MAX_RTR_SOLICITATION_DELAY. Since delaying DAD can result in significant delays in configuring a new care-of address when the Mobile Node moves to a new link, the Mobile Node preferably SHOULD NOT delay DAD when configuring a new care-of address. The Mobile Node SHOULD delay according to the mechanisms specified in RFC 2462 unless the implementation has a behavior that desynchronizes the steps that happen before the DAD in the case that multiple nodes experience handover at the same time. Such desynchronizing behaviors might be due to random delays in the L2 protocols or device drivers, or due to the movement detection mechanism that is used.

11.5.3 Using Multiple Care-of Addresses

As described in Section 11.5.2, a mobile node MAY use more than one care-of address at a time. Particularly in the case of many wireless networks, a mobile node effectively might be reachable through multiple links at the same time (e.g., with overlapping wireless cells), on which different on-link subnet prefixes may exist. The mobile node MUST ensure that its primary care-of address always has a prefix that is considered on-link by its current default router, i.e., advertised by its current default router in a solicited Router Advertisement. After selecting a new primary care-of address, the mobile node MUST send a Binding Update containing that care-of

address to its home agent. The Binding Update **MUST** have the Home Registration (H) and Acknowledge (A) bits set its home agent, as described on Section 11.7.1.

To assist with smooth handovers, a mobile node **SHOULD** retain its previous primary care-of address as a (non-primary) care-of address, and **SHOULD** still accept packets at this address, even after registering its new primary care-of address with its home agent. This is reasonable, since the mobile node could only receive packets at its previous primary care-of address if it were indeed still connected to that link. If the previous primary care-of address was allocated using stateful Address Autoconfiguration [29], the mobile node may not wish to release the address immediately upon switching to a new primary care-of address.

Whenever a mobile node determines that it is no longer reachable through a given link, it **SHOULD** invalidate all care-of addresses associated with address prefixes that it discovered from routers on the unreachable link which are not in the current set of address prefixes advertised by the (possibly new) current default router.

11.5.4 Returning Home

A mobile node detects that it has returned to its home link through the movement detection algorithm in use (Section 11.5.1), when the mobile node detects that its home subnet prefix is again on-link. The mobile node **SHOULD** then send a Binding Update to its home agent, to instruct its home agent to no longer intercept or tunnel packets for it. In this home registration, the mobile node **MUST** set the Acknowledge (A) and Home Registration (H) bits, set the Lifetime field to zero, and set the care-of address for the binding to the mobile node's own home address. The mobile node **MUST** use its home address as the source address in the Binding Update.

When sending this Binding Update to its home agent, the mobile node must be careful in how it uses Neighbor Solicitation [12] (if needed) to learn the home agent's link-layer address, since the home agent will be currently configured to intercept packets to the mobile node's home address using Duplicate Address Detection (DAD). In particular, the mobile node is unable to use its home address as the Source Address in the Neighbor Solicitation until the home agent stops defending the home address.

Neighbor Solicitation by the mobile node for the home agent's address will normally not be necessary, since the mobile node has already learned the home agent's link-layer address from a Source Link-Layer Address option in a Router Advertisement. However, if there are multiple home agents it may still be necessary to send a

solicitation. In this special case of the mobile node returning home, the mobile node **MUST** multicast the packet, and in addition set the Source Address of this Neighbor Solicitation to the unspecified address (0:0:0:0:0:0:0:0). The target of the Neighbor Solicitation **MUST** be set to the mobile node's home address. The destination IP address **MUST** be set to the Solicited-Node multicast address [3]. The home agent will send a multicast Neighbor Advertisement back to the mobile node with the Solicited flag (S) set to zero. In any case, the mobile node **SHOULD** record the information from the Source Link-Layer Address option or from the advertisement, and set the state of the Neighbor Cache entry for the home agent to REACHABLE.

The mobile node then sends its Binding Update to the home agent's link-layer address, instructing its home agent to no longer serve as a home agent for it. By processing this Binding Update, the home agent will cease defending the mobile node's home address for Duplicate Address Detection and will no longer respond to Neighbor Solicitations for the mobile node's home address. The mobile node is then the only node on the link receiving packets at the mobile node's home address. In addition, when returning home prior to the expiration of a current binding for its home address, and configuring its home address on its network interface on its home link, the mobile node **MUST NOT** perform Duplicate Address Detection on its own home address, in order to avoid confusion or conflict with its home agent's use of the same address. This rule also applies to the derived link-local address of the mobile node, if the Link Local Address Compatibility (L) bit was set when the binding was created. If the mobile node returns home after the bindings for all of its care-of addresses have expired, then it **SHOULD** perform DAD.

After the Mobile Node sends the Binding Update, it **MUST** be prepared to reply to Neighbor Solicitations for its home address. Such replies **MUST** be sent using a unicast Neighbor Advertisement to the sender's link-layer address. It is necessary to reply, since sending the Binding Acknowledgement from the home agent may require performing Neighbor Discovery, and the mobile node may not be able to distinguish Neighbor Solicitations coming from the home agent from other Neighbor Solicitations. Note that a race condition exists where both the mobile node and the home agent respond to the same solicitations sent by other nodes; this will be only temporary, however, until the Binding Update is accepted.

After receiving the Binding Acknowledgement for its Binding Update to its home agent, the mobile node **MUST** multicast onto the home link (to the all-nodes multicast address) a Neighbor Advertisement [12], to advertise the mobile node's own link-layer address for its own home address. The Target Address in this Neighbor Advertisement **MUST** be set to the mobile node's home address, and the Advertisement **MUST**

include a Target Link-layer Address option specifying the mobile node's link-layer address. The mobile node **MUST** multicast such a Neighbor Advertisement for each of its home addresses, as defined by the current on-link prefixes, including its link-local address and site-local address. The Solicited Flag (S) in these Advertisements **MUST NOT** be set, since they were not solicited by any Neighbor Solicitation. The Override Flag (O) in these Advertisements **MUST** be set, indicating that the Advertisements **SHOULD** override any existing Neighbor Cache entries at any node receiving them.

Since multicasting on the local link (such as Ethernet) is typically not guaranteed to be reliable, the mobile node **MAY** retransmit these Neighbor Advertisements [12] up to `MAX_NEIGHBOR_ADVERTISEMENT` times to increase their reliability. It is still possible that some nodes on the home link will not receive any of these Neighbor Advertisements, but these nodes will eventually be able to recover through use of Neighbor Unreachability Detection [12].

11.6 Return Routability Procedure

This section defines the rules that the mobile node must follow when performing the return routability procedure. Section 11.7.2 describes the rules when the return routability procedure needs to be initiated.

11.6.1 Sending Test Init Messages

A mobile node that initiates a return routability procedure **MUST** send (in parallel) a Home Test Init message and a Care-of Test Init messages. However, if the mobile node has recently received (see Section 5.2.7) one or both home or care-of keygen tokens, and associated nonce indices for the desired addresses, it **MAY** reuse them. Therefore, the return routability procedure may in some cases be completed with only one message pair. It may even be completed without any messages at all, if the mobile node has a recent home keygen token and has previously visited the same care-of address so that it also has a recent care-of keygen token. If the mobile node intends to send a Binding Update with the Lifetime set to zero and the care-of address equal to its home address – such as when returning home – sending a Home Test Init message is sufficient. In this case, generation of the binding management key depends exclusively on the home keygen token (Section 5.2.5).

A Home Test Init message **MUST** be created as described in Section 6.1.3.

A Care-of Test Init message **MUST** be created as described in Section 6.1.4. When sending a Home Test Init or Care-of Test Init message

the mobile node MUST record in its Binding Update List the following fields from the messages:

- o The IP address of the node to which the message was sent.
- o The home address of the mobile node. This value will appear in the Source Address field of the Home Test Init message. When sending the Care-of Test Init message, this address does not appear in the message, but represents the home address for which the binding is desired.
- o The time at which each of these messages was sent.
- o The cookies used in the messages.

Note that a single Care-of Test Init message may be sufficient even when there are multiple home addresses. In this case the mobile node MAY record the same information in multiple Binding Update List entries.

11.6.2 Receiving Test Messages

Upon receiving a packet carrying a Home Test message, a mobile node MUST validate the packet according to the following tests:

- o The Source Address of the packet belongs to a correspondent node for which the mobile node has a Binding Update List entry with a state indicating that return routability procedure is in progress. Note that there may be multiple such entries.
- o The Binding Update List indicates that no home keygen token has been received yet.
- o The Destination Address of the packet has the home address of the mobile node, and the packet has been received in a tunnel from the home agent.
- o The Home Init Cookie field in the message matches the value stored in the Binding Update List.

Any Home Test message not satisfying all of these tests MUST be silently ignored. Otherwise, the mobile node MUST record the Home Nonce Index and home keygen token in the Binding Update List. If the Binding Update List entry does not have a care-of keygen token, the mobile node SHOULD continue waiting for the Care-of Test message.

Upon receiving a packet carrying a Care-of Test message, a mobile node MUST validate the packet according to the following tests:

- o The Source Address of the packet belongs to a correspondent node for which the mobile node has a Binding Update List entry with a state indicating that return routability procedure is in progress. Note that there may be multiple such entries.
- o The Binding Update List indicates that no care-of keygen token has been received yet.
- o The Destination Address of the packet is the current care-of address of the mobile node.
- o The Care-of Init Cookie field in the message matches the value stored in the Binding Update List.

Any Care-of Test message not satisfying all of these tests **MUST** be silently ignored. Otherwise, the mobile node **MUST** record the Care-of Nonce Index and care-of keygen token in the Binding Update List. If the Binding Update List entry does not have a home keygen token, the mobile node **SHOULD** continue waiting for the Home Test message.

If after receiving either the Home Test or the Care-of Test message and performing the above actions, the Binding Update List entry has both the home and the care-of keygen tokens, the return routability procedure is complete. The mobile node **SHOULD** then proceed with sending a Binding Update as described in Section 11.7.2.

Correspondent nodes from the time before this specification was published may not support the Mobility Header protocol. These nodes will respond to Home Test Init and Care-of Test Init messages with an ICMP Parameter Problem code 1. The mobile node **SHOULD** take such messages as an indication that the correspondent node cannot provide route optimization, and revert back to the use of bidirectional tunneling.

11.6.3 Protecting Return Routability Packets

The mobile node **MUST** support the protection of Home Test and Home Test Init messages as described in Section 10.4.6.

When IPsec is used to protect return routability signaling or payload packets, the mobile node **MUST** set the source address it uses for the outgoing tunnel packets to the current primary care-of address. The mobile node starts to use a new primary care-of address immediately after sending a Binding Update to the home agent to register this new address.

11.7 Processing Bindings

11.7.1 Sending Binding Updates to the Home Agent

After deciding to change its primary care-of address as described in Section 11.5.1 and Section 11.5.2, a mobile node **MUST** register this care-of address with its home agent in order to make this its primary care-of address.

Also, if the mobile node wants the services of the home agent beyond the current registration period, the mobile node should send a new Binding Update to it well before the expiration of this period, even if it is not changing its primary care-of address. However, if the home agent returned a Binding Acknowledgement for the current registration with Status field set to 1 (accepted but prefix discovery necessary), the mobile node should not try to register again before it has learned the validity of its home prefixes through mobile prefix discovery. This is typically necessary every time this Status value is received, because information learned earlier may have changed.

To register a care-of address or to extend the lifetime of an existing registration, the mobile node sends a packet to its home agent containing a Binding Update, with the packet constructed as follows:

- o The Home Registration (H) bit **MUST** be set in the Binding Update.
- o The Acknowledge (A) bit **MUST** be set in the Binding Update.
- o The packet **MUST** contain a Home Address destination option, giving the mobile node's home address for the binding.
- o The care-of address for the binding **MUST** be used as the Source Address in the packet's IPv6 header, unless an Alternate Care-of Address mobility option is included in the Binding Update. This option **MUST** be included in all home registrations, as the ESP protocol will not be able to protect care-of addresses in the IPv6 header. (Mobile IPv6 implementations that know they are using IPsec AH to protect a particular message might avoid this option. For brevity the usage of AH is not discussed in this document.)
- o If the mobile node's link-local address has the same interface identifier as the home address for which it is supplying a new care-of address, then the mobile node **SHOULD** set the Link-Local Address Compatibility (L) bit.
- o If the home address was generated using RFC 3041 [18], then the link local address is unlikely to have a compatible interface identifier. In this case, the mobile node **MUST** clear the

Link-Local Address Compatibility (L) bit.

- o If the IPsec security associations between the mobile node and the home agent have been established dynamically, and the mobile node has the capability to update its endpoint in the used key management protocol to the new care-of address every time it moves, the mobile node SHOULD set the Key Management Mobility Capability (K) bit in the Binding Update. Otherwise, the mobile node MUST clear the bit.
- o The value specified in the Lifetime field SHOULD be less than or equal to the remaining valid lifetime of the home address and the care-of address specified for the binding.

Mobile nodes that use dynamic home agent address discovery should be careful with long lifetimes. If the mobile node loses the knowledge of its binding with a specific home agent, registering a new binding with another home agent may be impossible as the previous home agent is still defending the existing binding. Therefore, mobile nodes that use home agent address discovery SHOULD ensure information about their bindings is not lost, de-register before losing this information, or use small lifetimes.

The Acknowledge (A) bit in the Binding Update requests the home agent to return a Binding Acknowledgement in response to this Binding Update. As described in Section 6.1.8, the mobile node SHOULD retransmit this Binding Update to its home agent until it receives a matching Binding Acknowledgement. Once reaching a retransmission timeout period of MAX_BINDACK_TIMEOUT, the mobile node SHOULD restart the process of delivering the Binding Update, but trying instead the next home agent returned during dynamic home agent address discovery (see Section 11.4.1). If there was only one home agent, the mobile node instead SHOULD continue to periodically retransmit the Binding Update at this rate until acknowledged (or until it begins attempting to register a different primary care-of address). See Section 11.8 for information about retransmitting Binding Updates.

With the Binding Update, the mobile node requests the home agent to serve as the home agent for the given home address. Until the lifetime of this registration expires, the home agent considers itself the home agent for this home address.

Each Binding Update MUST be authenticated as coming from the right mobile node, as defined in Section 5.1. The mobile node MUST use its home address – either in the Home Address destination option or in the Source Address field of the IPv6 header – in Binding Updates sent to the home agent. This is necessary in order to allow the IPsec

policies to be matched with the right home address.

When sending a Binding Update to its home agent, the mobile node **MUST** also create or update the corresponding Binding Update List entry, as specified in Section 11.7.2.

The last Sequence Number value sent to the home agent in a Binding Update is stored by the mobile node. If the sending mobile node has no knowledge of the right Sequence Number value, it may start at any value. If the home agent rejects the value, it sends back a Binding Acknowledgement with status code 135, and the last accepted sequence number in the Sequence Number field of the Binding Acknowledgement. The mobile node **MUST** store this information and use the next Sequence Number value for the next Binding Update it sends.

If the mobile node has additional home addresses, then the mobile node **SHOULD** send an additional packet containing a Binding Update to its home agent to register the care-of address for each such other home address.

The home agent will only perform DAD for the mobile node's home address when the mobile node has supplied a valid binding between its home address and a care-of address. If some time elapses during which the mobile node has no binding at the home agent, it might be possible for another node to autoconfigure the mobile node's home address. Therefore, the mobile node **MUST** treat creation of a new binding with the home agent using an existing home address the same as creation of a new home address. In the unlikely event that the mobile node's home address is autoconfigured as the IPv6 address of another network node on the home network, the home agent will reply to the mobile node's subsequent Binding Update with a Binding Acknowledgement containing a Status of 134 (Duplicate Address Detection failed). In this case, the mobile node **MUST NOT** attempt to re-use the same home address. It **SHOULD** continue to register care-of addresses for its other home addresses, if any. (Mechanisms outlined in Appendix B.5 may in the future allow mobile nodes to acquire new home addresses to replace the one for which Status 134 was received.)

11.7.2 Correspondent Registration

When the mobile node is assured that its home address is valid, it can initiate a correspondent registration with the purpose of allowing the correspondent node to cache the mobile node's current care-of address. This procedure consists of the return routability procedure followed by a registration.

This section defines when to initiate the correspondent registration, and rules to follow when performing it.

After the mobile node has sent a Binding Update to its home agent to register a new primary care-of address (as described in Section 11.7.1), the mobile node **SHOULD** initiate a correspondent registration for each node that already appears in the mobile node's Binding Update List. The initiated procedures can be used to either update or delete binding information in the correspondent node.

For nodes that do not appear in the mobile node's Binding Update List, the mobile node **MAY** initiate a correspondent registration at any time after sending the Binding Update to its home agent. Considerations regarding when (and if) to initiate the procedure depend on the specific movement and traffic patterns of the mobile node and are outside the scope of this document.

In addition, the mobile node **MAY** initiate the procedure in response to receiving a packet that meets all of the following tests:

- o The packet was tunneled using IPv6 encapsulation.
- o The Destination Address in the tunnel (outer) IPv6 header is equal to any of the mobile node's care-of addresses.
- o The Destination Address in the original (inner) IPv6 header is equal to one of the mobile node's home addresses.
- o The Source Address in the tunnel (outer) IPv6 header differs from the Source Address in the original (inner) IPv6 header.
- o The packet does not contain a Home Test, Home Test Init, Care-of Test, or Care-of Test Init message.

If a mobile node has multiple home addresses, it becomes important to select the right home address to use in the correspondent registration. The used home address **MUST** be the Destination Address of the original (inner) packet.

The peer address used in the procedure **MUST** be determined as follows:

- o If a Home Address destination option is present in the original (inner) packet, the address from this option is used.
- o Otherwise, the Source Address in the original (inner) IPv6 header of the packet is used.

Note that the validity of the original packet is checked before attempting to initiate a correspondent registration. For instance, if a Home Address destination option appeared in the original packet, then rules in Section 9.3.1 are followed.

A mobile node MAY also choose to keep its topological location private from certain correspondent nodes, and thus need not initiate the correspondent registration.

Upon successfully completing the return routability procedure, and after receiving a successful Binding Acknowledgement from the Home Agent, a Binding Update MAY be sent to the correspondent node.

In any Binding Update sent by a mobile node, the care-of address (either the Source Address in the packet's IPv6 header or the Care-of Address in the Alternate Care-of Address mobility option of the Binding Update) MUST be set to one of the care-of addresses currently in use by the mobile node or to the mobile node's home address. A mobile node MAY set the care-of address differently for sending Binding Updates to different correspondent nodes.

A mobile node MAY also send a Binding Update to such a correspondent node to instruct it to delete any existing binding for the mobile node from its Binding Cache, as described in Section 6.1.7. Even in this case a successful completion of the return routability procedure is required first.

If the care-of address is not set to the mobile node's home address, the Binding Update requests the correspondent node to create or update an entry for the mobile node in the correspondent node's Binding Cache. This is done in order to record a care-of address for use in sending future packets to the mobile node. In this case, the value specified in the Lifetime field sent in the Binding Update SHOULD be less than or equal to the remaining lifetime of the home registration and the care-of address specified for the binding. The care-of address given in the Binding Update MAY differ from the mobile node's primary care-of address.

If the Binding Update is sent to request the correspondent node to delete any existing Binding Cache entry that it has for the mobile node, the care-of address is set to the mobile node's home address and the Lifetime field set to zero. In this case, generation of the binding management key depends exclusively on the home keygen token (Section 5.2.5). The care-of nonce index SHOULD be set to zero in this case. In keeping with the Binding Update creation rules below, the care-of address MUST be set to the home address if the mobile node is at home, or to the current care-of address if it is away from home.

If the mobile node wants to ensure that its new care-of address has been entered into a correspondent node's Binding Cache, the mobile node needs to request an acknowledgement by setting the Acknowledge (A) bit in the Binding Update.

A Binding Update is created as follows:

- o The current care-of address of the mobile node **MUST** be sent either in the Source Address of the IPv6 header or in the Alternate Care-of Address mobility option.
- o The Destination Address of the IPv6 header **MUST** contain the address of the correspondent node.
- o The Mobility Header is constructed according to rules in Section 6.1.7 and Section 5.2.6, including the Binding Authorization Data (calculated as defined in Section 6.2.7) and possibly the Nonce Indices mobility options.
- o The home address of the mobile node **MUST** be added to the packet in a Home Address destination option, unless the Source Address is the home address.

Each Binding Update **MUST** have a Sequence Number greater than the Sequence Number value sent in the previous Binding Update to the same destination address (if any). The sequence numbers are compared modulo 2^{16} , as described in Section 9.5.1. There is no requirement, however, that the Sequence Number value strictly increase by 1 with each new Binding Update sent or received, as long as the value stays within the window. The last Sequence Number value sent to a destination in a Binding Update is stored by the mobile node in its Binding Update List entry for that destination. If the sending mobile node has no Binding Update List entry, the Sequence Number **SHOULD** start at a random value. The mobile node **MUST NOT** use the same Sequence Number in two different Binding Updates to the same correspondent node, even if the Binding Updates provide different care-of addresses.

The mobile node is responsible for the completion of the correspondent registration, as well as any retransmissions that may be needed (subject to the rate limiting defined in Section 11.8).

11.7.3 Receiving Binding Acknowledgements

Upon receiving a packet carrying a Binding Acknowledgement, a mobile node **MUST** validate the packet according to the following tests:

- o The packet meets the authentication requirements for Binding Acknowledgements, defined in Section 6.1.8 and Section 5. That is, if the Binding Update was sent to the home agent, underlying IPsec protection is used. If the Binding Update was sent to the correspondent node, the Binding Authorization Data mobility option **MUST** be present and have a valid value.

- o The Binding Authorization Data mobility option, if present, **MUST** be the last option and **MUST** not have trailing padding.
- o The Sequence Number field matches the Sequence Number sent by the mobile node to this destination address in an outstanding Binding Update.

Any Binding Acknowledgement not satisfying all of these tests **MUST** be silently ignored.

When a mobile node receives a packet carrying a valid Binding Acknowledgement, the mobile node **MUST** examine the Status field as follows:

- o If the Status field indicates that the Binding Update was accepted (the Status field is less than 128), then the mobile node **MUST** update the corresponding entry in its Binding Update List to indicate that the Binding Update has been acknowledged; the mobile node **MUST** then stop retransmitting the Binding Update. In addition, if the value specified in the Lifetime field in the Binding Acknowledgement is less than the Lifetime value sent in the Binding Update being acknowledged, then the mobile node **MUST** subtract the difference between these two Lifetime values from the remaining lifetime for the binding as maintained in the corresponding Binding Update List entry (with a minimum value for the Binding Update List entry lifetime of 0). That is, if the Lifetime value sent in the Binding Update was L_{update} , the Lifetime value received in the Binding Acknowledgement was L_{ack} , and the current remaining lifetime of the Binding Update List entry is L_{remain} , then the new value for the remaining lifetime of the Binding Update List entry should be

$$\max((L_{remain} - (L_{update} - L_{ack})), 0)$$

where $\max(X, Y)$ is the maximum of X and Y . The effect of this step is to correctly manage the mobile node's view of the binding's remaining lifetime (as maintained in the corresponding Binding Update List entry) so that it correctly counts down from the Lifetime value given in the Binding Acknowledgement, but with the timer countdown beginning at the time that the Binding Update was sent.

Mobile nodes **SHOULD** send a new Binding Update well before the expiration of this period in order to extend the lifetime. This helps to avoid disruptions in communications, which might otherwise be caused by network delays or clock drift.

- o Additionally, if the Status field value is 1 (Accepted but prefix discovery necessary), the mobile node SHOULD send a Mobile Prefix Solicitation message to update its information about the available prefixes.
- o If the Status field indicates that the Binding Update was rejected (the Status field is greater than or equal to 128), then the mobile node can take steps to correct the cause of the error and retransmit the Binding Update (with a new Sequence Number value), subject to the rate limiting restriction specified in Section 11.8. If this is not done, or it fails, then the mobile node SHOULD record in its Binding Update List that future Binding Updates SHOULD NOT be sent to this destination.

The treatment of a Binding Refresh Advice mobility option within the Binding Acknowledgement depends on the where the acknowledgement came from. This option MUST be ignored if the acknowledgement came from a correspondent node. If it came from the home agent, the mobile node uses Refresh Interval field in the option as a suggestion that it SHOULD attempt to refresh its home registration at the indicated shorter interval.

If the acknowledgement came from the home agent, the mobile node examines the value of the Key Management Mobility Capability (K) bit. If this bit is not set, the mobile node SHOULD discard key management protocol connections, if any, to the home agent. The mobile node MAY also initiate a new key management connection.

If this bit is set, the mobile node SHOULD move its own endpoint in the key management protocol connections to the home agent, if any. The mobile node's new endpoint should be the new care-of address. For an IKE phase 1 connection, this means packets sent to this address with the original ISAKMP cookies are accepted.

11.7.4 Receiving Binding Refresh Requests

When a mobile node receives a packet containing a Binding Refresh Request message, the mobile node has a Binding Update List entry for the source of the Binding Refresh Request, and the mobile node wants to retain its binding cache entry at the correspondent node, then the mobile node should start a return routability procedure. If the mobile node wants to have its binding cache entry removed it can either ignore the Binding Refresh Request and wait for the binding to time out, or it can at any time delete its binding from a correspondent node with an explicit binding update with zero lifetime and the care-of address set to the home address. If the mobile node does not know if it needs the binding cache entry, it can make the decision in an implementation dependent manner, such as based on

available resources.

Note that the mobile node should be careful to not respond to Binding Refresh Requests for addresses not in the Binding Update List to avoid being subjected to a denial of service attack.

If the return routability procedure completes successfully, a Binding Update message SHOULD be sent as described in Section 11.7.2. The Lifetime field in this Binding Update SHOULD be set to a new lifetime, extending any current lifetime remaining from a previous Binding Update sent to this node (as indicated in any existing Binding Update List entry for this node), and lifetime SHOULD again be less than or equal to the remaining lifetime of the home registration and the care-of address specified for the binding. When sending this Binding Update, the mobile node MUST update its Binding Update List in the same way as for any other Binding Update sent by the mobile node.

11.8 Retransmissions and Rate Limiting

The mobile node is responsible for retransmissions and rate limiting in the return routability procedure, registrations, and in solicited prefix discovery.

When the mobile node sends a Mobile Prefix Solicitation, Home Test Init, Care-of Test Init or Binding Update for which it expects a response, the mobile node has to determine a value for the initial retransmission timer:

- o If the mobile node is sending a Mobile Prefix Solicitation, it SHOULD use an initial retransmission interval of INITIAL_SOLICIT_TIMER (see Section 12).
- o If the mobile node is sending a Binding Update and it does not have an existing binding at the home agent, it SHOULD use InitialBindackTimeoutFirstReg (see Section 13) as a value for the initial retransmission timer. This long retransmission interval will allow the home agent to complete the Duplicate Address Detection procedure which is mandated in this case, as detailed in Section 11.7.1.
- o Otherwise, the mobile node should use the specified value of INITIAL_BINDACK_TIMEOUT for the initial retransmission timer.

If the mobile node fails to receive a valid, matching response within the selected initial retransmission interval, the mobile node SHOULD retransmit the message, until a response is received.

The retransmissions by the mobile node MUST use an exponential back-off process, in which the timeout period is doubled upon each retransmission until either the node receives a response or the timeout period reaches the value `MAX_BINDACK_TIMEOUT`. The mobile node MAY continue to send these messages at this slower rate indefinitely.

The mobile node SHOULD start a separate back-off process for different message types, different home addresses and different care-of addresses. However, in addition an overall rate limitation applies for messages sent to a particular correspondent node. This ensures that the correspondent node has sufficient amount of time to answer when bindings for multiple home addresses are registered, for instance. The mobile node MUST NOT send Mobility Header messages of a particular type to a particular correspondent node more than `MAX_UPDATE_RATE` times within a second.

Retransmitted Binding Updates MUST use a Sequence Number value greater than that used for the previous transmission of this Binding Update. Retransmitted Home Test Init and Care-of Test Init messages MUST use new cookie values.

12. Protocol Constants

DHAAD_RETRIES	4 retransmissions
INITIAL_BINDACK_TIMEOUT	1 second
INITIAL_DHAAD_TIMEOUT	3 seconds
INITIAL_SOLICIT_TIMER	3 seconds
MAX_BINDACK_TIMEOUT	32 seconds
MAX_NONCE_LIFETIME	240 seconds
MAX_TOKEN_LIFETIME	210 seconds
MAX_RR_BINDING_LIFETIME	420 seconds
MAX_UPDATE_RATE	3 times
PREFIX_ADV_RETRIES	3 retransmissions
PREFIX_ADV_TIMEOUT	3 seconds

13. Protocol Configuration Variables

MaxMobPfxAdvInterval	Default: 86,400 seconds
MinDelayBetweenRAs	Default: 3 seconds, Min: 0.03 seconds
MinMobPfxAdvInterval	Default: 600 seconds
InitialBindackTimeoutFirstReg	Default: 1.5 seconds

Home agents **MUST** allow the first three variables to be configured by system management, and mobile nodes **MUST** allow the last variable to be configured by system management.

The default value for InitialBindackTimeoutFirstReg has been calculated as 1.5 times the default value of RetransTimer [12] times the default value of DupAddrDetectTransmits [13].

The value MinDelayBetweenRAs overrides the value of the protocol constant MIN_DELAY_BETWEEN_RAS, as specified in RFC 2461 [12]. This variable **SHOULD** be set to MinRtrAdvInterval, if MinRtrAdvInterval is less than 3 seconds.

14. IANA Considerations

This document defines a new IPv6 protocol, the Mobility Header, described in Section 6.1. This protocol must be assigned a protocol number.

This document also creates a new name space "Mobility Header Type", for the MH Type field in the Mobility Header. The current message types are described starting from Section 6.1.2, and are the following:

- 0 Binding Refresh Request
- 1 Home Test Init
- 2 Care-of Test Init
- 3 Home Test
- 4 Care-of Test
- 5 Binding Update
- 6 Binding Acknowledgement
- 7 Binding Error

Future values of the MH Type can be allocated using standards action [10].

Furthermore, each mobility message may contain mobility options as described in Section 6.2. This document defines a new name space "Mobility Option" to identify these options. The current mobility options are defined starting from Section 6.2.2 and are the following:

- 0 Pad1
- 1 PadN
- 2 Binding Refresh Advice
- 3 Alternate Care-of Address
- 4 Nonce Indices

5 Authorization Data

Future values of the Option Type can be allocated using standards action [10].

This document also defines a new IPv6 destination option, the Home Address option, described in Section 6.3. This option has already been assigned the Option Type value 0xC9.

This document also defines a new IPv6 type 2 routing header, described in Section 6.4. The value 2 is to be allocated by IANA when this specification becomes an RFC.

In addition, this document defines four ICMP message types, two used as part of the dynamic home agent address discovery mechanism and two used in lieu of Router Solicitations and Advertisements when the mobile node is away from the home link. These messages must be assigned ICMPv6 type numbers from the informational message range:

- o The Home Agent Address Discovery Request message, described in Section 6.5;
- o The Home Agent Address Discovery Reply message, described in Section 6.6;
- o The Mobile Prefix Solicitation, described in Section 6.7; and
- o The Mobile Prefix Advertisement, described in Section 6.8.

This document also defines two new Neighbor Discovery [12] options, which must be assigned Option Type values within the option numbering space for Neighbor Discovery messages:

- o The Advertisement Interval option, described in Section 7.3; and
- o The Home Agent Information option, described in Section 7.4.

15. Security Considerations

15.1 Threats

Any mobility solution must protect itself against misuses of the mobility features and mechanisms. In Mobile IPv6, most of the potential threats are concerned with false Bindings, usually resulting in Denial-of-Service attacks. Some of the threats also pose potential for Man-in-the-Middle, Hijacking, Confidentiality, and Impersonation attacks. The main threats this protocol protects against are the following:

- o Threats involving Binding Updates sent to home agents and correspondent nodes. For instance, an attacker might claim that a certain mobile node is currently at a different location than it really is. If a home agent accepts such spoofed information sent to it, the mobile node might not get traffic destined to it. Similarly, a malicious (mobile) node might use the home address of a victim node in a forged Binding Update sent to a correspondent node.

These pose threats against confidentiality, integrity, and availability. That is, an attacker might learn the contents of packets destined to another node by redirecting the traffic to itself. Furthermore, an attacker might use the redirected packets in an attempt to set itself as a Man-in-the-Middle between a mobile and a correspondent node. This would allow the attacker to impersonate the mobile node, leading to integrity and availability problems.

A malicious (mobile) node might also send Binding Updates in which the care-of address is set to the address of a victim node. If such Binding Updates were accepted, the malicious node could lure the correspondent node into sending potentially large amounts of data to the victim; the correspondent node's replies to messages sent by the malicious mobile node will be sent to the victim host or network. This could be used to cause a Distributed Denial-of-Service attack. For example, the correspondent node might be a site that will send a high-bandwidth stream of video to anyone who asks for it. Note that the use of flow-control protocols such as TCP does not necessarily defend against this type of attack, because the attacker can fake the acknowledgements. Even keeping TCP initial sequence numbers secret does not help, because the attacker can receive the first few segments (including the ISN) at its own address, and only then redirect the stream to the victim's address. These types of attacks may also be directed to networks instead of nodes. Further variations of this threat are described elsewhere

[27, 34].

An attacker might also attempt to disrupt a mobile node's communications by replaying a Binding Update that the node had sent earlier. If the old Binding Update was accepted, packets destined for the mobile node would be sent to its old location and not its current location.

In conclusion, there are Denial-of-Service, Man-in-the-Middle, Confidentiality, and Impersonation threats against the parties involved in sending legitimate Binding Updates, and Denial-of-Service threats against any other party.

- o Threats associated with payload packets: Payload packets exchanged with mobile nodes are exposed to similar threats as regular IPv6 traffic is. However, Mobile IPv6 introduces the Home Address destination option, a new routing header type (type 2), and uses tunneling headers in the payload packets. The protocol must protect against potential new threats involving the use of these mechanisms.

Third parties become exposed to a reflection threat via the Home Address destination option, unless appropriate security precautions are followed. The Home Address destination option could be used to direct response traffic toward a node whose IP address appears in the option. In this case, ingress filtering would not catch the forged "return address" [36, 32].

A similar threat exists with the tunnels between the mobile node and the home agent. An attacker might forge tunnel packets between the mobile node and the home agent, making it appear that the traffic is coming from the mobile node when it is not. Note that an attacker who is able to forge tunnel packets would typically be able to forge also packets that appear to come directly from the mobile node. This is not a new threat as such. However, it may make it easier for attackers to escape detection by avoiding ingress filtering and packet tracing mechanisms. Furthermore, spoofed tunnel packets might be used to gain access to the home network.

Finally, a routing header could also be used in reflection attacks, and in attacks designed to bypass firewalls. The generality of the regular routing header would allow circumvention of IP-address based rules in firewalls. It would also allow reflection of traffic to other nodes. These threats exist with routing headers in general, even if the usage that Mobile IPv6 requires is safe.

- o Threats associated with dynamic home agent and mobile prefix discovery.
- o Threats against the Mobile IPv6 security mechanisms themselves: An attacker might, for instance, lure the participants into executing expensive cryptographic operations or allocating memory for the purpose of keeping state. The victim node would have no resources left to handle other tasks.

As a fundamental service in an IPv6 stack, Mobile IPv6 is expected to be deployed in most nodes of the IPv6 Internet. The above threats should therefore be considered in the light of being applicable to the whole Internet.

It should also be noted that some additional threats result from movements as such, even without the involvement of mobility protocols. Mobile nodes must be capable to defend themselves in the networks that they visit, as typical perimeter defenses applied in the home network no longer protect them.

15.2 Features

This specification provides a series of features designed to mitigate the risk introduced by the threats listed above. The main security features are the following:

- o Reverse Tunneling as a mandatory feature.
- o Protection of Binding Updates sent to home agents.
- o Protection of Binding Updates sent to correspondent nodes.
- o Protection against reflection attacks that use the Home Address destination option.
- o Protection of tunnels between the mobile node and the home agent.
- o Closing routing header vulnerabilities.
- o Mitigating Denial-of-Service threats to the Mobile IPv6 security mechanisms themselves.

The support for encrypted reverse tunneling (see Section 11.3.1) allows mobile nodes to defeat certain kinds of traffic analysis.

Protecting those Binding Updates that are sent to home agents and those that are sent to arbitrary correspondent nodes requires very different security solutions due to the different situations. Mobile

nodes and home agents are expected to be naturally subject to the network administration of the home domain.

Thus, they can and are supposed to have a security association that can be used to reliably authenticate the exchanged messages. See Section 5.1 for the description of the protocol mechanisms, and Section 15.3 below for a discussion of the resulting level of security.

It is expected that Mobile IPv6 route optimization will be used on a global basis between nodes belonging to different administrative domains. It would be a very demanding task to build an authentication infrastructure on this scale. Furthermore, a traditional authentication infrastructure cannot be easily used to authenticate IP addresses, because IP addresses can change often. It is not sufficient to just authenticate the mobile nodes. Authorization to claim the right to use an address is needed as well. Thus, an "infrastructureless" approach is necessary. The chosen infrastructureless method is described in Section 5.2 and Section 15.4 discusses the resulting security level and the design rationale of this approach.

Specific rules guide the use of the Home Address destination option, the routing header, and the tunneling headers in the payload packets. These rules are necessary to remove the vulnerabilities associated with their unrestricted use. The effect of the rules is discussed in Section 15.7, Section 15.8, and Section 15.9.

Denial-of-Service threats against Mobile IPv6 security mechanisms themselves concern mainly the Binding Update procedures with correspondent nodes. The protocol has been designed to limit the effects of such attacks, as will be described in Section 15.4.5.

15.3 Binding Updates to Home Agent

Signaling between the mobile node and the home agent requires message integrity. This is necessary to assure the home agent that a Binding Update is from a legitimate mobile node. In addition, correct ordering and anti-replay protection are optionally needed.

IPsec ESP protects the integrity of the Binding Updates and Binding Acknowledgements, by securing mobility messages between the mobile node and the home agent.

IPsec can provide anti-replay protection only if dynamic keying is used (which may not always be the case). IPsec also does not guarantee correct ordering of packets, only that they have not been replayed. Because of this, sequence numbers within the Mobile IPv6

messages are used to ensure correct ordering (see Section 5.1). However, if the 16 bit Mobile IPv6 sequence number space is cycled through, or the home agent reboots and loses its state regarding the sequence numbers, replay and reordering attacks become possible. The use of dynamic keying, IPsec anti-replay protection, and the Mobile IPv6 sequence numbers can together prevent such attacks. It is also recommended that use of non-volatile storage is considered for home agents, to avoid losing their state.

A sliding window scheme is used for the sequence numbers. The protection against replays and reordering attacks without a key management mechanism works when the attacker remembers up to a maximum of 2^{15} Binding Updates.

The above mechanisms do not show that the care-of address given in the Binding Update is correct. This opens the possibility for Denial-of-Service attacks against third parties. However, since the mobile node and home agent have a security association, the home agent can always identify an ill-behaving mobile node. This allows the home agent operator to discontinue the mobile node's service, and possibly take further actions based on the business relationship with the mobile node's owner.

Note that the use of a single pair of manually keyed security associations conflicts with the generation of a new home addresses [18] for the mobile node, or with the adoption of a new home subnet prefix. This is because IPsec security associations are bound to the used addresses. While certificate-based automatic keying alleviates this problem to an extent, it is still necessary to ensure that a given mobile node cannot send Binding Updates for the address of another mobile node. In general, this leads to the inclusion of home addresses in certificates in the Subject AltName field. This again limits the introduction of new addresses without either manual or automatic procedures to establish new certificates. Therefore, this specification restricts the generation of new home addresses (for any reason) to those situations where there already exists a security association or certificate for the new address. (Appendix B.4 lists the improvement of security for new addresses as one of the future developments for Mobile IPv6.)

Support for IKE has been specified as optional. The following should be observed about the use of manual keying:

- o As discussed above, with manually keyed IPsec only a limited form of protection exists against replay and reordering attacks. A vulnerability exists if either the sequence number space is cycled through, or if the home agent reboots and forgets its sequence numbers (and uses volatile memory to store the sequence numbers).

Assuming the mobile node moves continuously every 10 minutes, it takes roughly 455 days before the sequence number space has been cycled through. Typical movement patterns rarely reach this high frequency today.

- o A mobile node and its home agent belong to the same domain. If this were not the case, manual keying would not be possible [28], but in Mobile IPv6 only these two parties need to know the manually configured keys. Similarly, we note that Mobile IPv6 employs standard block ciphers in IPsec, and is not vulnerable to problems associated with stream ciphers and manual keying.
- o It is expected that the owner of the mobile node and the administrator of the home agent agree on the used keys and other parameters with some off-line mechanism.

The use of IKEv1 with Mobile IPv6 is documented in more detail in [21]. The following should be observed from the use of IKEv1:

- o It is necessary to prevent a mobile node from claiming another mobile node's home address. The home agent must verify that the mobile node trying to negotiate the SA for a particular home address is authorized for that home address. This implies that even with the use of IKE, a policy entry needs to be configured for each home address served by the home agent.

It may be possible to include home addresses in the Subject AltName field of certificate to avoid this. However, implementations are not guaranteed to support the use of a particular IP address (care-of address) while another address (home address) appears in the certificate. In any case, even this approach would require user-specific tasks in the certificate authority.

- o If preshared secret authentication is used, IKEv1 main mode cannot be used. Aggressive mode or group preshared secrets need to be used instead, with corresponding security implications.

Note that like many other issues, this is a general IKEv1 issue related to the ability to use different IP addresses, and not specifically related to Mobile IPv6. For further information, see Section 4.4 in [21].

- o Due to the problems outlined in Section 11.3.2, IKE phase 1 between the mobile node and its home agent is established using the mobile node's current care-of address. This implies that when the mobile node moves to a new location, it may have to re-establish phase 1. A Key Management Mobility Capability (K) flag is provided for implementations that can update the IKE phase 1 endpoints without re-establishing phase 1, but the support for

this behavior is optional.

- o When certificates are used, IKE fragmentation can occur as discussed in Section 7 in [21].
- o Nevertheless, even if per-mobile node configuration is required even with IKE, an important benefit of IKE is that it automates the negotiation of cryptographic parameters, including the SPIs, cryptographic algorithms, and so on. Thus less configuration information is needed.
- o The frequency of movements in some link layers or deployment scenarios may be high enough to make replay and reordering attacks possible, if only manual keying is used. IKE SHOULD be used in such cases. Potentially vulnerable scenarios involve continuous movement through small cells, or uncontrolled alternation between available network attachment points.
- o Similarly, in some deployment scenarios the number of mobile nodes may be very large. In these cases it can be necessary to use automatic mechanisms to reduce the management effort in the administration of cryptographic parameters, even if some per-mobile node configuration is always needed. IKE SHOULD also be used in such cases.
- o Other automatic key management mechanisms exist beyond IKEv1, but this document does not address the issues related to them. We note, however, that most of the above discussion applies to IKEv2 [30] as well, at least as it is currently specified.

15.4 Binding Updates to Correspondent Nodes

The motivation for designing the return routability procedure was to have sufficient support for Mobile IPv6, without creating significant new security problems. The goal for this procedure was not to protect against attacks that were already possible before the introduction of Mobile IPv6.

The next sections will describe the security properties of the used method, both from the point of view of possible on-path attackers who can see those cryptographic values that have been sent in the clear (Section 15.4.2 and Section 15.4.3) or from the point of view of other attackers (Section 15.4.6).

15.4.1 Overview

The chosen infrastructureless method verifies that the mobile node is

"live" (that is, it responds to probes) at its home and care-of addresses. Section 5.2 describes the return routability procedure in detail. The procedure uses the following principles:

- o A message exchange verifies that the mobile node is reachable at its addresses i.e. is at least able to transmit and receive traffic at both the home and care-of addresses.
- o The eventual Binding Update is cryptographically bound to the tokens supplied in the exchanged messages.
- o Symmetric exchanges are employed to avoid the use of this protocol in reflection attacks. In a symmetric exchange, the responses are always sent to the same address as the request was sent from.
- o The correspondent node operates in a stateless manner until it receives a fully authorized Binding Update.
- o Some additional protection is provided by encrypting the tunnels between the mobile node and home agent with IPsec ESP. As the tunnel transports also the nonce exchanges, this limits the ability of attackers to see these nonces. For instance, this prevents attacks launched from the mobile node's current foreign link even when no link-layer confidentiality is available.

The resulting level of security is in theory the same even without this additional protection: the return routability tokens are still exposed only to one path within the whole Internet.

However, the mobile nodes are often found on an insecure link, such as a public access Wireless LAN. Thus this addition makes a practical difference in many cases.

For further information about the design rationale of the return routability procedure, see [27, 34, 33, 32]. The used mechanisms have been adopted from these documents.

15.4.2 Achieved Security Properties

The return routability procedure protects Binding Updates against all attackers who are unable to monitor the path between the home agent and the correspondent node. The procedure does not defend against attackers who can monitor this path. Note that such attackers are in any case able to mount an active attack against the mobile node when it is at its home location. The possibility of such attacks is not an impediment to the deployment of Mobile IPv6, because these attacks are possible regardless of whether Mobile IPv6 is in use.

This procedure also protects against Denial-of-Service attacks in

which the attacker pretends to be a mobile, but uses the victim's address as the care-of address. This would cause the correspondent node to send the victim some unexpected traffic. The procedure defends against these attacks by requiring at least passive presence of the attacker at the care-of address or on the path from the correspondent to the care-of address. Normally, this will be the mobile node.

15.4.3 Comparison to Regular IPv6 Communications

This section discusses the protection offered by the return routability method by comparing it to the security of regular IPv6 communications. We will divide vulnerabilities in three classes: (1) those related to attackers on the local network of the mobile node, home agent, or the correspondent node, (2) those related to attackers on the path between the home network and the correspondent node, and (3) off-path attackers, i.e. the rest of the Internet.

We will now discuss the vulnerabilities of regular IPv6 communications. The on-link vulnerabilities of IPv6 communications include Denial-of-Service, Masquerading, Man-in-the-Middle, Eavesdropping, and other attacks. These attacks can be launched through spoofing Router Discovery, Neighbor Discovery and other IPv6 mechanisms. Some of these attacks can be prevented with the use of cryptographic protection in the packets.

A similar situation exists with on-path attackers. That is, without cryptographic protection the traffic is completely vulnerable.

Assuming that attackers have not penetrated the security of the Internet routing protocols, attacks are much harder to launch from off-path locations. Attacks that can be launched from these locations are mainly Denial-of-Service attacks, such as flooding and/or reflection attacks. It is not possible for an off-path attacker to become a Man-in-the-Middle.

Next, we will consider the vulnerabilities that exist when IPv6 is used together with Mobile IPv6 and the return routability procedure. On the local link the vulnerabilities are same as those as in IPv6, but Masquerade and Man-in-the-Middle attacks can now be launched also against future communications, and not just against current communications. If a Binding Update was sent while the attacker was present on the link, its effects stay during the lifetime of the binding. This happens even if the attacker moves away from the link. In contrast, an attacker who uses only plain IPv6 generally has to stay on the link in order to continue the attack. Note that in order to launch these new attacks, the IP address of the victim must be known. This makes this attack feasible mainly in the context of

well-known interface IDs, such as those already appearing in the traffic on the link or registered in the DNS.

On-path attackers can exploit similar vulnerabilities as in regular IPv6. There are some minor differences, however. Masquerade, Man-in-the-Middle, and Denial-of-Service attacks can be launched with just the interception of a few packets, whereas in regular IPv6 it is necessary to intercept every packet. The effect of the attacks is the same regardless of the method, however. In any case, the most difficult task attacker faces in these attacks is getting on the right path.

The vulnerabilities for off-path attackers are the same as in regular IPv6. Those nodes that are not on the path between the home agent and the correspondent node will not be able to receive the home address probe messages.

In conclusion, we can state the following main results from this comparison:

- o Return routability procedure prevents any off-path attacks beyond those that are already possible in regular IPv6. This is the most important result, and prevents attackers from the Internet from exploiting any vulnerabilities.
- o Vulnerabilities to attackers on the home agent link, the correspondent node link, and the path between them are roughly the same as in regular IPv6.
- o However, one difference is that in basic IPv6 an on-path attacker must be constantly present on the link or the path, whereas with Mobile IPv6 an attacker can leave a binding behind after moving away.

For this reason, this specification limits the creation of bindings to at most `MAX_TOKEN_LIFETIME` seconds after the last routability check has been performed, and limits the duration of a binding to at most `MAX_RR_BINDING_LIFETIME` seconds. With these limitation, attackers cannot take practical advantages of this vulnerability.

- o There are some other minor differences, such as an effect to the Denial-of-Service vulnerabilities. These can be considered to be insignificant.
- o The path between the home agent and a correspondent node is typically easiest to attack on the links at either end, in particular if these links are publicly accessible wireless LANs.

Attacks against the routers or switches on the path are typically harder to accomplish. The security on layer 2 of the links plays then a major role in the resulting overall network security. Similarly, security of IPv6 Neighbor and Router Discovery on these links has a large impact. If these were secured using some new technology in the future, this could change the situation regarding the easiest point of attack.

For a more in-depth discussion of these issues, see [32].

15.4.4 Replay Attacks

The return routability procedure also protects the participants against replayed Binding Updates. The attacker is unable to replay the same message due to the sequence number which is a part of the Binding Update. It is also unable to modify the Binding Update since the MAC verification would fail after such a modification.

Care must be taken when removing bindings at the correspondent node, however. If a binding is removed while the nonce used in its creation is still valid, an attacker could replay the old Binding Update. Rules outlined in Section 5.2.8 ensure that this cannot happen.

15.4.5 Denial-of-Service Attacks

The return routability procedure has protection against resource exhaustion Denial-of-Service attacks. The correspondent nodes do not retain any state about individual mobile nodes until an authentic Binding Update arrives. This is achieved through the construct of keygen tokens from the nonces and node keys that are not specific to individual mobile nodes. The keygen tokens can be reconstructed by the correspondent node, based on the home and care-of address information that arrives with the Binding Update. This means that the correspondent nodes are safe against memory exhaustion attacks except where on-path attackers are concerned. Due to the use of symmetric cryptography, the correspondent nodes are relatively safe against CPU resource exhaustion attacks as well.

Nevertheless, as [27] describes, there are situations in which it is impossible for the mobile and correspondent nodes to determine if they actually need a binding or whether they just have been fooled into believing so by an attacker. Therefore, it is necessary to consider situations where such attacks are being made.

Even if route optimization is a very important optimization, it is still only an optimization. A mobile node can communicate with a correspondent node even if the correspondent refuses to accept any

Binding Updates. However, performance will suffer because packets from the correspondent node to the mobile node will be routed via the mobile's home agent rather than a more direct route. A correspondent node can protect itself against some of these resource exhaustion attacks as follows. If the correspondent node is flooded with a large number of Binding Updates that fail the cryptographic integrity checks, it can stop processing Binding Updates. If a correspondent node finds that it is spending more resources on checking bogus Binding Updates than it is likely to save by accepting genuine Binding Updates, then it may silently discard some or all Binding Updates without performing any cryptographic operations.

Layers above IP can usually provide additional information to decide if there is a need to establish a binding with a specific peer. For example, TCP knows if the node has a queue of data that it is trying to send to a peer. An implementation of this specification is not required to make use of information from higher protocol layers, but some implementations are likely to be able to manage resources more effectively by making use of such information.

We also require that all implementations be capable of administratively disabling route optimization.

15.4.6 Key Lengths

Attackers can try to break the return routability procedure in many ways. Section 15.4.2 discusses the situation where the attacker can see the cryptographic values sent in the clear, and Section 15.4.3 discusses the impact this has on IPv6 communications. This section discusses whether attackers can guess the right values without seeing them.

While the return routability procedure is in progress, 64 bit cookies are used to protect spoofed responses. This is believed to be sufficient, given that to blindly spoof a response a very large number of messages would have to be sent before success would be probable.

The tokens used in the return routability procedure provide together 128 bits of information. This information is used internally as an input to a hash function to produce a 160 bit quantity suitable for producing the keyed hash in the Binding Update using the HMAC_SHA1 algorithm. The final keyed hash length is 96 bits. The limiting factors in this case are the input token lengths and the final keyed hash length. The internal hash function application does not reduce the entropy.

The 96 bit final keyed hash is of typical size and believed to be

secure. The 128 bit input from the tokens is broken in two pieces, the home keygen token and the care-of keygen token. An attacker can try to guess the right cookie value, but again this would require a large number of messages, in the average 2^{63} messages for one or 2^{127} for two. Furthermore, given that the cookies are valid only for a short period of time, the attack has to keep a high constant message rate to achieve a lasting effect. This does not appear practical.

When the mobile node is returning home, it is allowed to use just the home keygen token of 64 bits. This is less than 128 bits, but attacking it blindly would still require a large number of messages to be sent. If the attacker is on the path and capable of seeing the Binding Update, it could conceivably break the keyed hash with brute force. However, in this case the attacker has to be on path, which appears to offer easier ways for denial-of-service than preventing route optimization.

15.5 Dynamic Home Agent Address Discovery

The dynamic home agent address discovery function could be used to learn the addresses of home agents in the home network.

The ability to learn addresses of nodes may be useful to attackers, because brute-force scanning of the address space is not practical with IPv6. Thus, they could benefit from any means which make mapping the networks easier. For example, if a security threat targeted at routers or even home agents is discovered, having a simple ICMP mechanism to find out possible targets easily may prove to be an additional (though minor) security risk.

Apart from discovering the address(es) of home agents, attackers will not be able to learn much from this information, however, and mobile nodes cannot be tricked into using wrong home agents as all other communication with the home agents is secure.

15.6 Mobile Prefix Discovery

The mobile prefix discovery function may leak interesting information about network topology and prefix lifetimes to eavesdroppers, and for this reason requests for this information have to be authenticated. Responses and unsolicited prefix information needs to be authenticated to prevent the mobile nodes from being tricked into believing false information about the prefixes, and possibly preventing communications with the existing addresses. Optionally, encryption may be applied to prevent leakage of the prefix information.

15.7 Tunneling via the Home Agent

Tunnels between the mobile node and the home agent can be protected by ensuring proper use of source addresses, and optional cryptographic protection. These procedures are discussed in Section 5.5.

Binding Updates to the home agents are secure. When receiving tunneled traffic the home agent verifies the outer IP address corresponds to the current location of the mobile node. This acts as a weak form of protection against spoofing packets that appear to come from the mobile node. This is particularly useful, if no end-to-end security is being applied between the mobile and correspondent nodes. The outer IP address check prevents attacks where the attacker is controlled by ingress filtering. It also prevents attacks when the attacker does not know the current care-of address of the mobile node. Attackers who know the care-of address and are not controlled by ingress filtering could still send traffic through the home agent. This includes attackers on the same local link as the mobile node is currently on. But such attackers could in any case send packets that appear to come from the mobile node, without attacking the tunnel; the attacker could simply send packets with the source address set to the mobile node's home address. However, this attack does not work if the final destination of the packet is in the home network, and some form of perimeter defense is being applied for packets sent to those destinations. In such cases it is recommended that either end-to-end security or additional tunnel protection is applied, as is usual in remote access situations.

Home agents and mobile nodes may use IPsec ESP to protect payload packets tunneled between themselves. This is useful to protect communications against attackers on the path of the tunnel.

When site local home address are used, reverse tunneling can be used to send site local traffic from another location. Administrators should be aware of this when allowing such home addresses. In particular, the outer IP address check described above is not sufficient against all attackers. The use of encrypted tunnels is particularly useful for this kind of home addresses.

15.8 Home Address Option

When the mobile node sends packets directly to the correspondent node, the Source Address field of the packet's IPv6 header is the care-of address. Ingress filtering [26] works therefore in the usual manner even for mobile nodes, as the Source Address is topologically correct. The Home Address option is used to inform the correspondent

node of the mobile node's home address.

However, the care-of address in the Source Address field does not survive in replies sent by the correspondent node unless it has a binding for this mobile node. Also, not all attacker tracing mechanisms work when packets are being reflected through correspondent nodes using the Home Address option. For these reasons, this specification restricts the use of the Home Address option. It may only be used when a binding has already been established with the participation of the node at the home address, as described in Section 5.5 and Section 6.3. This prevents reflection attacks through the use of the Home Address option. It also ensures that the correspondent nodes reply to the same address as the mobile node sends traffic from.

No special authentication of the Home Address option is required beyond the above, but note that if the IPv6 header of a packet is covered by IPsec Authentication Header, then that authentication covers the Home Address option as well. Thus, even when authentication is used in the IPv6 header, the security of the Source Address field in the IPv6 header is not compromised by the presence of a Home Address option. Without authentication of the packet, then any field in the IPv6 header, including the Source Address field, and any other parts of the packet, including the Home Address option, can be forged or modified in transit. In this case, the contents of the Home Address option is no more suspect than any other part of the packet.

15.9 Type 2 Routing Header

The definition of the type 2 routing header is described in Section 6.4. This definition and the associated processing rules have been chosen so that the header cannot be used for what is traditionally viewed as source routing. In particular, the Home Address in the routing header will always have to be assigned to the home address of the receiving node. Otherwise the packet will be dropped.

Generally, source routing has a number of security concerns. These include the automatic reversal of unauthenticated source routes (which is an issue for IPv4, but not for IPv6). Another concern is the ability to use source routing to "jump" between nodes inside, as well as outside a firewall. These security concerns are not issues in Mobile IPv6, due to the rules mentioned above.

In essence the semantics of the type 2 routing header is the same as a special form of IP-in-IP tunneling where the inner and outer source addresses are the same.

* * *

This implies that a device which implements filtering of packets should be able to distinguish between a type 2 routing header and other routing headers, as required in Section 8.3. This is necessary in order to allow Mobile IPv6 traffic while still having the option to filter out other uses of routing headers.

16. Contributors

Tuomas Aura, Mike Roe, Greg O'Shea, Pekka Nikander, Erik Nordmark, and Michael Thomas worked on the return routability protocols which eventually led to the procedures used in this protocol. The procedures described in [34] were adopted in the protocol.

Significant contributions were made by members of the Mobile IPv6 Security Design Team, including (in alphabetical order) Gabriel Montenegro, Erik Nordmark and Pekka Nikander, who have contributed volumes of text to this specification.

17. Acknowledgements

We would like to thank the members of the Mobile IP and IPng Working Groups for their comments and suggestions on this work. We would particularly like to thank (in alphabetical order) Fred Baker, Josh Broch, Samita Chakrabarti, Robert Chalmers, Noel Chiappa, Greg Daley, Vijay Devarapalli, Rich Draves, Francis Dupont, Thomas Eklund, Jun-ichiro Itojun Hagino, Brian Haley, Marc Hasson, John Ioannidis, James Kempf, Rajeev Koodli, Krishna Kumar, T.J. Kniveton, Joe Lau, Jiwoong Lee, Aime Le Rouzic, Vesa-Matti Mantyla, Kevin Miles, Glenn Morrow, Thomas Narten, Karen Nielsen, Simon Nybroe, David Oran, Brett Pentland, Lars Henrik Petander, Basavaraj Patil, Mohan Parthasarathy, Alexandru Petrescu, Mattias Petterson, Ken Powell, Phil Roberts, Ed Rimmell, Patrice Romand, Luis A. Sanchez, Jeff Schiller, Pekka Savola, Arvind Sevalkar, Keiichi Shima, Tom Soderlund, Hesham Soliman, Jim Solomon, Tapio Suihko, Dave Thaler, Benny Van Houdt, Jon-Olov Vatn, Carl E. Williams, Vladislav Yasevich, Alper Yegin, and Xinhua Zhao, for their detailed reviews of earlier versions of this document. Their suggestions have helped to improve both the design and presentation of the protocol.

We would also like to thank the participants of the Mobile IPv6 testing event (1999), implementors who participated Mobile IPv6 interoperability testing at Connectathons (2000, 2001, 2002, and 2003), and the participants at the ETSI interoperability testing (2000, 2002). Finally, we would like to thank the TAHI project who has provided test suites for Mobile IPv6.

Normative References

- [1] Eastlake, D., Crocker, S. and J. Schiller, "Randomness Recommendations for Security", RFC 1750, December 1994.
- [2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [3] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.
- [4] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [5] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.
- [6] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998.
- [7] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.
- [8] Maughan, D., Schertler, M., Schneider, M. and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.
- [9] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.
- [10] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.
- [11] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.
- [12] Narten, T., Nordmark, E. and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [13] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.
- [14] Conta, A. and S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", RFC 2463, December 1998.
- [15] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6

Specification", RFC 2473, December 1998.

- [16] Johnson, D. and S. Deering, "Reserved IPv6 Subnet Anycast Addresses", RFC 2526, March 1999.
- [17] Deering, S., Fenner, W. and B. Haberman, "Multicast Listener Discovery (MLD) for IPv6", RFC 2710, October 1999.
- [18] Narten, T. and R. Draves, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 3041, January 2001.
- [19] Reynolds, J., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, January 2002.
- [20] National Institute of Standards and Technology, "Secure Hash Standard", FIPS PUB 180-1, April 1995, <<http://www.itl.nist.gov/fipspubs/fip180-1.htm>>.
- [21] Arkko, J., Devarapalli, V. and F. Dupont, "Using IPsec to Protect Mobile IPv6 Signaling between Mobile Nodes and Home Agents", draft-ietf-mobileip-mipv6-ha-ipsec-06 (work in progress), June 2003.

Informative References

- [22] Perkins, C., "IP Mobility Support", RFC 2002, October 1996.
- [23] Perkins, C., "IP Encapsulation within IP", RFC 2003, October 1996.
- [24] Perkins, C., "Minimal Encapsulation within IP", RFC 2004, October 1996.
- [25] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.
- [26] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", RFC 2267, January 1998.
- [27] Aura, T. and J. Arkko, "MIPv6 BU Attacks and Defenses", draft-aura-mipv6-bu-attacks-01 (work in progress), March 2002.
- [28] Bellovin, S., "Guidelines for Mandating Automated Key Management", draft-bellovin-mandate-keymgmt-00 (work in progress), August 2003.
- [29] Droms, R., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", draft-ietf-dhc-dhcpv6-28 (work in progress), November 2002.
- [30] Kaufman, C., "Internet Key Exchange (IKEv2) Protocol", draft-ietf-ipsec-ikev2-07 (work in progress), April 2003.
- [31] Draves, R., "Default Address Selection for IPv6", draft-ietf-ipv6-default-addr-select-09 (work in progress), August 2002.
- [32] Nikander, P., Aura, T., Arkko, J., Montenegro, G. and E. Nordmark, "Mobile IP version 6 Route Optimization Security Design Background", draft-nikander-mobileip-v6-ro-sec-00.txt (work in progress), April 2003.
- [33] Nordmark, E., "Securing MIPv6 BUs using return routability (BU3WAY)", draft-nordmark-mobileip-bu3way-00 (work in progress), November 2001.
- [34] Roe, M., Aura, T., O'Shea, G. and J. Arkko, "Authentication of Mobile IPv6 Binding Updates and Acknowledgments", draft-roe-mobileip-updateauth-02 (work in progress), March 2002.

- [35] Savola, P., "Use of /127 Prefix Length Between Routers Considered Harmful", draft-savola-ipv6-127-prefixlen-04 (work in progress), June 2002.
- [36] Savola, P., "Security of IPv6 Routing Header and Home Address Options", draft-savola-ipv6-rh-ha-security-03 (work in progress), December 2002.
- [37] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", draft-vida-mld-v2-06 (work in progress), December 2002.

Authors' Addresses

David B. Johnson
Rice University
Dept. of Computer Science, MS 132
6100 Main Street
Houston TX 77005-1892
USA

EMail: dbj@cs.rice.edu

Charles E. Perkins
Nokia Research Center
313 Fairchild Drive
Mountain View CA 94043
USA

EMail: charliep@iprg.nokia.com

Jari Arkko
Ericsson
Jorvas 02420
Finland

EMail: jari.arkko@ericsson.com

Appendix A. Changes from Previous Version of the Draft

This appendix briefly lists some of the major changes in this draft relative to the previous version of this same draft, draft-ietf-mobileip-ipv6-23.txt:

- o A typo in Section 6.8 has been corrected: rules regarding reception of Mobile Prefix Advertisements apply to mobile nodes, not home agents (tracked issue 317).
- o A sentence has been removed from Section 10.3.1, since it dealt with the tracking of router advertisements by other home agents, which is a function that was already removed earlier in an IESG review (tracked issue 316).

Appendix B. Future Extensions

B.1 Piggybacking

This document does not specify how to piggyback payload packets on the binding related messages. However, it is envisioned that this can be specified in a separate document when currently discussed issues such as the interaction between piggybacking and IPsec are fully resolved (see also Appendix B.3). The return routability messages can indicate support for piggybacking with a new mobility option.

B.2 Triangular Routing

Due to the concerns about opening reflection attacks with the Home Address destination option, this specification requires that this option must be verified against the Binding Cache, i.e., there must be a Binding Cache entry for the Home Address and Care-of Address.

Future extensions may be specified that allow the use of unverified Home Address destination options in ways that do not introduce security issues.

B.3 New Authorization Methods

While the return routability procedure provides a good level of security, there exists methods that have even higher levels of security. Secondly, as discussed in Section 15.4, future enhancements of IPv6 security may cause a need to improve also the security of the return routability procedure. Using IPsec as the sole method for authorizing Binding Updates to correspondent nodes is also possible. The protection of the Mobility Header for this purpose is easy, though one must ensure that the IPsec SA was created with appropriate authorization to use the home address referenced in the Binding Update. For instance, a certificate used by IKE to create the security association might contain the home address. A future specification may specify how this is done.

B.4 Dynamically Generated Home Addresses

A future version of this specification may include functionality that allows the generation of new home addresses without requiring pre-arranged security associations or certificates even for the new addresses.

B.5 Remote Home Address Configuration

The method for initializing a mobile node's home addresses on

power-up or after an extended period of being disconnected from the network is beyond the scope of this specification. Whatever procedure is used should result in the mobile node having the same stateless or stateful (e.g., DHCPv6) home address autoconfiguration information it would have if it were attached to the home network. Due to the possibility that the home network could be renumbered while the mobile node is disconnected, a robust mobile node would not rely solely on storing these addresses locally.

Such a mobile node could initialize by using the following procedure:

1. Generate a care-of address.
2. Query DNS for an anycast address associated with the FQDN of the home agent(s).
3. Perform home agent address discovery, and select a home agent.
4. Configure one home address based on the selected home agent's subnet prefix and the interface identifier of the mobile node.
5. Create security associations and security policy database entries for protecting the traffic between the selected home address and home agent.
6. Perform a home registration to the selected home agent.
7. Perform mobile prefix discovery.
8. Make a decision if further home addresses need to be configured.

This procedure is restricted to those situations where the home prefix is 64 bits and the mobile node knows its own interface identifier of also 64 bits.

B.6 Neighbor Discovery Extensions

Future specifications may improve the efficiency of Neighbor Discovery tasks, which could be helpful for fast movements. One factor which is currently being looked at is the delays caused by the Duplicate Address Detection mechanism. Currently, Duplicate Address Detection needs to be performed for every new care-of address as the mobile node moves, and for the mobile node's link-local address on every new link. In particular, the need and the trade-offs of re-performing Duplicate Address Detection for the link-local address every time when the mobile node moves on to new links will need to be examined. Improvements in this area are, however, generally applicable and progressed independently from Mobile IPv6

specification.

Future functional improvements may also be relevant for Mobile IPv6 and other applications. For instance, mechanisms that would allow recovery from a Duplicate Address Detection collision would be useful for link-local, care-of, and home addresses.

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION

HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF
MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the
Internet Society.

Network Working Group
Request for Comments: 2460
Obsoletes: 1883
Category: Standards Track

S. Deering
Cisco
R. Hinden
Nokia
December 1998

Internet Protocol, Version 6 (IPv6) Specification

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This document specifies version 6 of the Internet Protocol (IPv6), also sometimes referred to as IP Next Generation or IPng.

Table of Contents

1. Introduction.....	2
2. Terminology.....	3
3. IPv6 Header Format.....	4
4. IPv6 Extension Headers.....	6
4.1 Extension Header Order.....	7
4.2 Options.....	9
4.3 Hop-by-Hop Options Header.....	11
4.4 Routing Header.....	12
4.5 Fragment Header.....	18
4.6 Destination Options Header.....	23
4.7 No Next Header.....	24
5. Packet Size Issues.....	24
6. Flow Labels.....	25
7. Traffic Classes.....	25
8. Upper-Layer Protocol Issues.....	27
8.1 Upper-Layer Checksums.....	27
8.2 Maximum Packet Lifetime.....	28
8.3 Maximum Upper-Layer Payload Size.....	28
8.4 Responding to Packets Carrying Routing Headers.....	29

Appendix A. Semantics and Usage of the Flow Label Field.....	30
Appendix B. Formatting Guidelines for Options.....	32
Security Considerations.....	35
Acknowledgments.....	35
Authors' Addresses.....	35
References.....	35
Changes Since RFC-1883.....	36
Full Copyright Statement.....	39

1. Introduction

IP version 6 (IPv6) is a new version of the Internet Protocol, designed as the successor to IP version 4 (IPv4) [RFC-791]. The changes from IPv4 to IPv6 fall primarily into the following categories:

o Expanded Addressing Capabilities

IPv6 increases the IP address size from 32 bits to 128 bits, to support more levels of addressing hierarchy, a much greater number of addressable nodes, and simpler auto-configuration of addresses. The scalability of multicast routing is improved by adding a "scope" field to multicast addresses. And a new type of address called an "anycast address" is defined, used to send a packet to any one of a group of nodes.

o Header Format Simplification

Some IPv4 header fields have been dropped or made optional, to reduce the common-case processing cost of packet handling and to limit the bandwidth cost of the IPv6 header.

o Improved Support for Extensions and Options

Changes in the way IP header options are encoded allows for more efficient forwarding, less stringent limits on the length of options, and greater flexibility for introducing new options in the future.

o Flow Labeling Capability

A new capability is added to enable the labeling of packets belonging to particular traffic "flows" for which the sender requests special handling, such as non-default quality of service or "real-time" service.

o Authentication and Privacy Capabilities

Extensions to support authentication, data integrity, and (optional) data confidentiality are specified for IPv6.

This document specifies the basic IPv6 header and the initially-defined IPv6 extension headers and options. It also discusses packet size issues, the semantics of flow labels and traffic classes, and the effects of IPv6 on upper-layer protocols. The format and semantics of IPv6 addresses are specified separately in [ADDRARCH]. The IPv6 version of ICMP, which all IPv6 implementations are required to include, is specified in [ICMPv6].

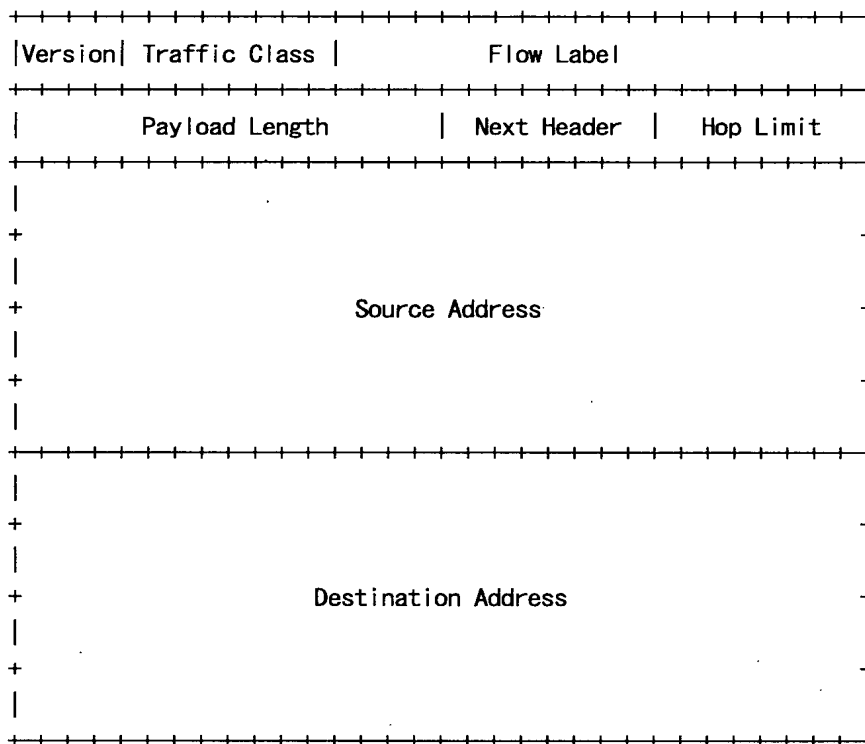
2. Terminology

- node - a device that implements IPv6.
- router - a node that forwards IPv6 packets not explicitly addressed to itself. [See Note below].
- host - any node that is not a router. [See Note below].
- upper layer - a protocol layer immediately above IPv6. Examples are transport protocols such as TCP and UDP, control protocols such as ICMP, routing protocols such as OSPF, and internet or lower-layer protocols being "tunneled" over (i.e., encapsulated in) IPv6 such as IPX, AppleTalk, or IPv6 itself.
- link - a communication facility or medium over which nodes can communicate at the link layer, i.e., the layer immediately below IPv6. Examples are Ethernet (simple or bridged); PPP links; X.25, Frame Relay, or ATM networks; and internet (or higher) layer "tunnels", such as tunnels over IPv4 or IPv6 itself.
- neighbors - nodes attached to the same link.
- interface - a node's attachment to a link.
- address - an IPv6-layer identifier for an interface or a set of interfaces.
- packet - an IPv6 header plus payload.
- link MTU - the maximum transmission unit, i.e., maximum packet size in octets, that can be conveyed over a link.

path MTU - the minimum link MTU of all the links in a path between a source node and a destination node.

Note: it is possible, though unusual, for a device with multiple interfaces to be configured to forward non-self-destined packets arriving from some set (fewer than all) of its interfaces, and to discard non-self-destined packets arriving from its other interfaces. Such a device must obey the protocol requirements for routers when receiving packets from, and interacting with neighbors over, the former (forwarding) interfaces. It must obey the protocol requirements for hosts when receiving packets from, and interacting with neighbors over, the latter (non-forwarding) interfaces.

3. IPv6 Header Format



```
Version      4-bit Internet Protocol version number = 6.
```

Traffic Class 8-bit traffic class field. See section 7.

Flow Label	20-bit flow label. See section 6.
------------	-----------------------------------

Payload Length	16-bit unsigned integer. Length of the IPv6 payload, i.e., the rest of the packet following this IPv6 header, in octets. (Note that any
----------------	---

extension headers [section 4] present are considered part of the payload, i.e., included in the length count.)

Next Header	8-bit selector. Identifies the type of header immediately following the IPv6 header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
Hop Limit	8-bit unsigned integer. Decrement by 1 by each node that forwards the packet. The packet is discarded if Hop Limit is decremented to zero.
Source Address	128-bit address of the originator of the packet. See [ADDRARCH].
Destination Address	128-bit address of the intended recipient of the packet (possibly not the ultimate recipient, if a Routing header is present). See [ADDRARCH] and section 4.4.

4. IPv6 Extension Headers

In IPv6, optional internet-layer information is encoded in separate headers that may be placed between the IPv6 header and the upper-layer header in a packet. There are a small number of such extension headers, each identified by a distinct Next Header value. As illustrated in these examples, an IPv6 packet may carry zero, one, or more extension headers, each identified by the Next Header field of the preceding header:

IPv6 header	TCP header + data
Next Header = TCP	

IPv6 header	Routing header	TCP header + data
Next Header = Routing	Next Header = TCP	

IPv6 header	Routing header	Fragment header	fragment of TCP header + data
Next Header = Routing	Next Header = Fragment	Next Header = TCP	

With one exception, extension headers are not examined or processed by any node along a packet's delivery path, until the packet reaches the node (or each of the set of nodes, in the case of multicast) identified in the Destination Address field of the IPv6 header. There, normal demultiplexing on the Next Header field of the IPv6 header invokes the module to process the first extension header, or the upper-layer header if no extension header is present. The contents and semantics of each extension header determine whether or not to proceed to the next header. Therefore, extension headers must be processed strictly in the order they appear in the packet; a receiver must not, for example, scan through a packet looking for a particular kind of extension header and process that header prior to processing all preceding ones.

The exception referred to in the preceding paragraph is the Hop-by-Hop Options header, which carries information that must be examined and processed by every node along a packet's delivery path, including the source and destination nodes. The Hop-by-Hop Options header, when present, must immediately follow the IPv6 header. Its presence is indicated by the value zero in the Next Header field of the IPv6 header.

If, as a result of processing a header, a node is required to proceed to the next header but the Next Header value in the current header is unrecognized by the node, it should discard the packet and send an ICMP Parameter Problem message to the source of the packet, with an ICMP Code value of 1 ("unrecognized Next Header type encountered") and the ICMP Pointer field containing the offset of the unrecognized value within the original packet. The same action should be taken if a node encounters a Next Header value of zero in any header other than an IPv6 header.

Each extension header is an integer multiple of 8 octets long, in order to retain 8-octet alignment for subsequent headers. Multi-octet fields within each extension header are aligned on their natural boundaries, i.e., fields of width n octets are placed at an integer multiple of n octets from the start of the header, for $n = 1, 2, 4$, or 8 .

A full implementation of IPv6 includes implementation of the following extension headers:

- Hop-by-Hop Options
- Routing (Type 0)
- Fragment
- Destination Options
- Authentication
- Encapsulating Security Payload

The first four are specified in this document; the last two are specified in [RFC-2402] and [RFC-2406], respectively.

4.1 Extension Header Order

When more than one extension header is used in the same packet, it is recommended that those headers appear in the following order:

- IPv6 header
- Hop-by-Hop Options header
- Destination Options header (note 1)
- Routing header
- Fragment header

Authentication header (note 2)
Encapsulating Security Payload header (note 2)
Destination Options header (note 3)
upper-layer header

note 1: for options to be processed by the first destination that appears in the IPv6 Destination Address field plus subsequent destinations listed in the Routing header.

note 2: additional recommendations regarding the relative order of the Authentication and Encapsulating Security Payload headers are given in [RFC-2406].

note 3: for options to be processed only by the final destination of the packet.

Each extension header should occur at most once, except for the Destination Options header which should occur at most twice (once before a Routing header and once before the upper-layer header).

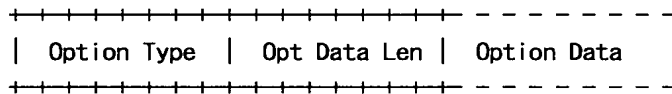
If the upper-layer header is another IPv6 header (in the case of IPv6 being tunneled over or encapsulated in IPv6), it may be followed by its own extension headers, which are separately subject to the same ordering recommendations.

If and when other extension headers are defined, their ordering constraints relative to the above listed headers must be specified.

IPv6 nodes must accept and attempt to process extension headers in any order and occurring any number of times in the same packet, except for the Hop-by-Hop Options header which is restricted to appear immediately after an IPv6 header only. Nonetheless, it is strongly advised that sources of IPv6 packets adhere to the above recommended order until and unless subsequent specifications revise that recommendation.

4.2 Options

Two of the currently-defined extension headers — the Hop-by-Hop Options header and the Destination Options header — carry a variable number of type-length-value (TLV) encoded "options", of the following format:



Option Type	8-bit identifier of the type of option.
Opt Data Len	8-bit unsigned integer. Length of the Option Data field of this option, in octets.
Option Data	Variable-length field. Option-Type-specific data.

The sequence of options within a header must be processed strictly in the order they appear in the header; a receiver must not, for example, scan through the header looking for a particular kind of option and process that option prior to processing all preceding ones.

The Option Type identifiers are internally encoded such that their highest-order two bits specify the action that must be taken if the processing IPv6 node does not recognize the Option Type:

- 00 - skip over this option and continue processing the header.
- 01 - discard the packet.
- 10 - discard the packet and, regardless of whether or not the packet's Destination Address was a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.
- 11 - discard the packet and, only if the packet's Destination Address was not a multicast address, send an ICMP Parameter Problem, Code 2, message to the packet's Source Address, pointing to the unrecognized Option Type.

The third-highest-order bit of the Option Type specifies whether or not the Option Data of that option can change en-route to the packet's final destination. When an Authentication header is present

in the packet, for any option whose data may change en-route, its entire Option Data field must be treated as zero-valued octets when computing or verifying the packet's authenticating value.

0 - Option Data does not change en-route

1 - Option Data may change en-route

The three high-order bits described above are to be treated as part of the Option Type, not independent of the Option Type. That is, a particular option is identified by a full 8-bit Option Type, not just the low-order 5 bits of an Option Type.

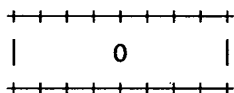
The same Option Type numbering space is used for both the Hop-by-Hop Options header and the Destination Options header. However, the specification of a particular option may restrict its use to only one of those two headers.

Individual options may have specific alignment requirements, to ensure that multi-octet values within Option Data fields fall on natural boundaries. The alignment requirement of an option is specified using the notation $xn+y$, meaning the Option Type must appear at an integer multiple of x octets from the start of the header, plus y octets. For example:

$2n$ means any 2-octet offset from the start of the header.
 $8n+2$ means any 8-octet offset from the start of the header, plus 2 octets.

There are two padding options which are used when necessary to align subsequent options and to pad out the containing header to a multiple of 8 octets in length. These padding options must be recognized by all IPv6 implementations:

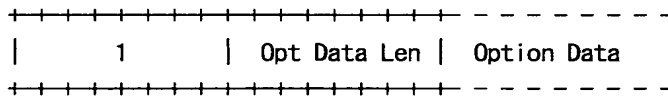
Pad1 option (alignment requirement: none)



NOTE! the format of the Pad1 option is a special case — it does not have length and value fields.

The Pad1 option is used to insert one octet of padding into the Options area of a header. If more than one octet of padding is required, the PadN option, described next, should be used, rather than multiple Pad1 options.

PadN option (alignment requirement: none)

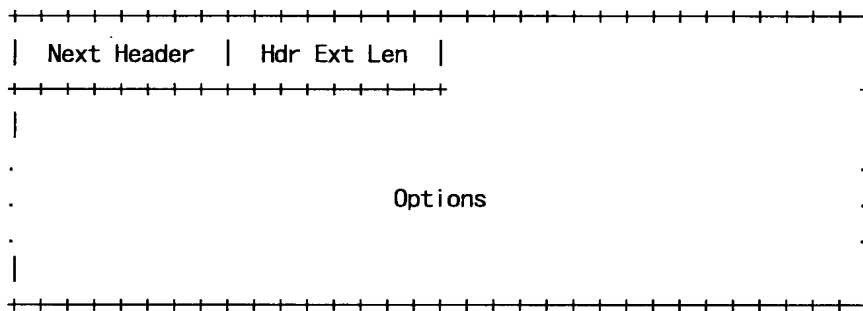


The PadN option is used to insert two or more octets of padding into the Options area of a header. For N octets of padding, the Opt Data Len field contains the value N-2, and the Option Data consists of N-2 zero-valued octets.

Appendix B contains formatting guidelines for designing new options.

4.3 Hop-by-Hop Options Header

The Hop-by-Hop Options header is used to carry optional information that must be examined by every node along a packet's delivery path. The Hop-by-Hop Options header is identified by a Next Header value of 0 in the IPv6 header, and has the following format:

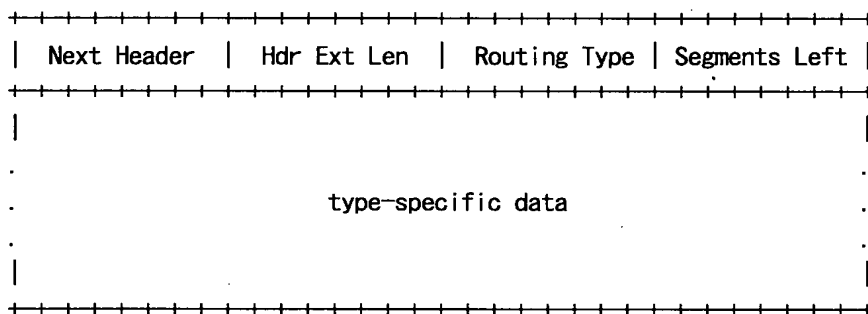


Next Header	8-bit selector. Identifies the type of header immediately following the Hop-by-Hop Options header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
Hdr Ext Len	8-bit unsigned integer. Length of the Hop-by-Hop Options header in 8-octet units, not including the first 8 octets.
Options	Variable-length field, of length such that the complete Hop-by-Hop Options header is an integer multiple of 8 octets long. Contains one or more TLV-encoded options, as described in section 4.2.

The only hop-by-hop options defined in this document are the Pad1 and PadN options specified in section 4.2.

4.4 Routing Header

The Routing header is used by an IPv6 source to list one or more intermediate nodes to be "visited" on the way to a packet's destination. This function is very similar to IPv4's Loose Source and Record Route option. The Routing header is identified by a Next Header value of 43 in the immediately preceding header, and has the following format:



Next Header	8-bit selector. Identifies the type of header immediately following the Routing header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
Hdr Ext Len	8-bit unsigned integer. Length of the Routing header in 8-octet units, not including the first 8 octets.
Routing Type	8-bit identifier of a particular Routing header variant.
Segments Left	8-bit unsigned integer. Number of route segments remaining, i.e., number of explicitly listed intermediate nodes still to be visited before reaching the final destination.
type-specific data	Variable-length field, of format determined by the Routing Type, and of length such that the complete Routing header is an integer multiple of 8 octets long.

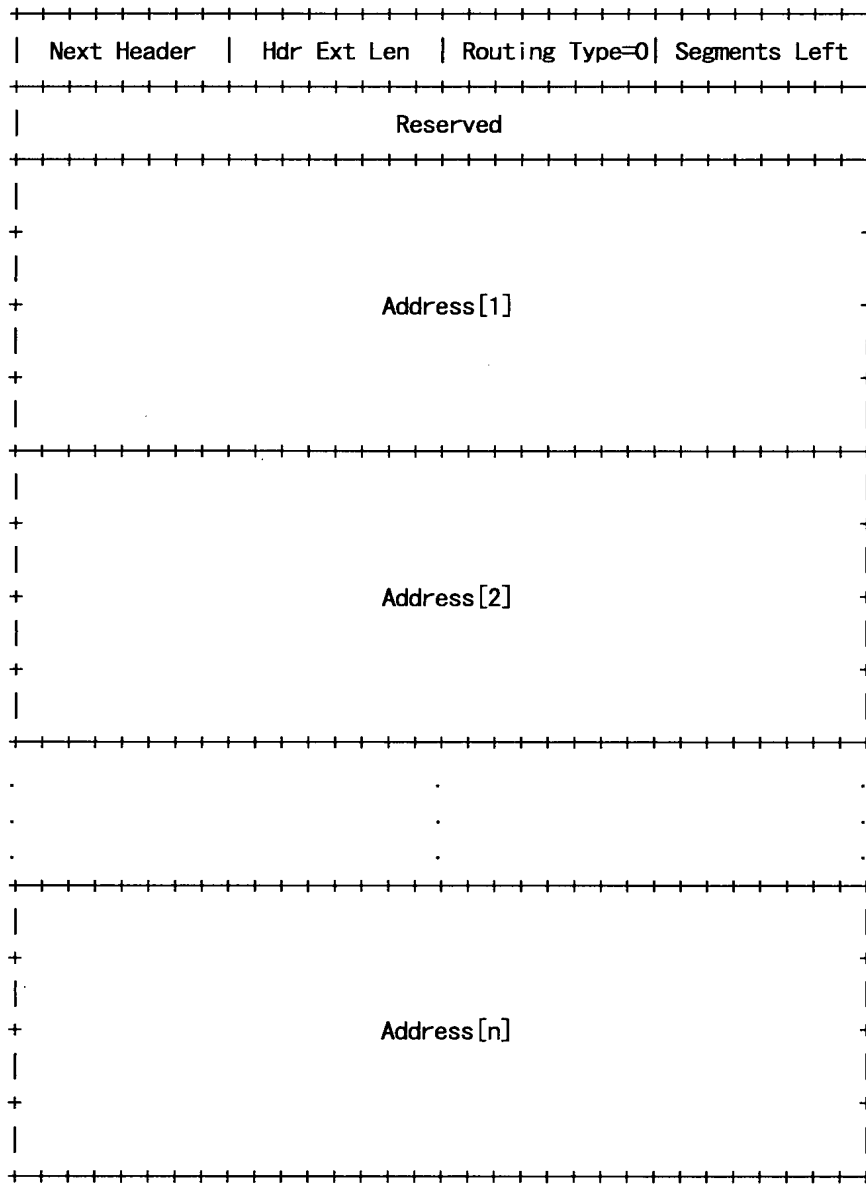
If, while processing a received packet, a node encounters a Routing header with an unrecognized Routing Type value, the required behavior of the node depends on the value of the Segments Left field, as follows:

If Segments Left is zero, the node must ignore the Routing header and proceed to process the next header in the packet, whose type is identified by the Next Header field in the Routing header.

If Segments Left is non-zero, the node must discard the packet and send an ICMP Parameter Problem, Code 0, message to the packet's Source Address, pointing to the unrecognized Routing Type.

If, after processing a Routing header of a received packet, an intermediate node determines that the packet is to be forwarded onto a link whose link MTU is less than the size of the packet, the node must discard the packet and send an ICMP Packet Too Big message to the packet's Source Address.

The Type 0 Routing header has the following format:



Next Header	8-bit selector. Identifies the type of header immediately following the Routing header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
Hdr Ext Len	8-bit unsigned integer. Length of the Routing header in 8-octet units, not including the first 8 octets. For the Type 0 Routing header, Hdr Ext Len is equal to two times the number of addresses in the header.
Routing Type	0.

Segments Left	8-bit unsigned integer. Number of route segments remaining, i.e., number of explicitly listed intermediate nodes still to be visited before reaching the final destination.
Reserved	32-bit reserved field. Initialized to zero for transmission; ignored on reception.
Address[1..n]	Vector of 128-bit addresses, numbered 1 to n.

Multicast addresses must not appear in a Routing header of Type 0, or in the IPv6 Destination Address field of a packet carrying a Routing header of Type 0.

A Routing header is not examined or processed until it reaches the node identified in the Destination Address field of the IPv6 header. In that node, dispatching on the Next Header field of the immediately preceding header causes the Routing header module to be invoked, which, in the case of Routing Type 0, performs the following algorithm:

```
if Segments Left = 0 {
    proceed to process the next header in the packet, whose type is
    identified by the Next Header field in the Routing header
}
else if Hdr Ext Len is odd {
    send an ICMP Parameter Problem, Code 0, message to the Source
    Address, pointing to the Hdr Ext Len field, and discard the
    packet
}
else {
    compute n, the number of addresses in the Routing header, by
    dividing Hdr Ext Len by 2

    if Segments Left is greater than n {
        send an ICMP Parameter Problem, Code 0, message to the Source
        Address, pointing to the Segments Left field, and discard the
        packet
    }
    else {
        decrement Segments Left by 1;
        compute i, the index of the next address to be visited in
        the address vector, by subtracting Segments Left from n

        if Address [i] or the IPv6 Destination Address is multicast {
            discard the packet
        }
        else {
            swap the IPv6 Destination Address and Address[i]

            if the IPv6 Hop Limit is less than or equal to 1 {
                send an ICMP Time Exceeded — Hop Limit Exceeded in
                Transit message to the Source Address and discard the
                packet
            }
            else {
                decrement the Hop Limit by 1

                resubmit the packet to the IPv6 module for transmission
                to the new destination
            }
        }
    }
}
```

As an example of the effects of the above algorithm, consider the case of a source node S sending a packet to destination node D, using a Routing header to cause the packet to be routed via intermediate nodes I1, I2, and I3. The values of the relevant IPv6 header and Routing header fields on each segment of the delivery path would be as follows:

As the packet travels from S to I1:

Source Address = S	Hdr Ext Len = 6
Destination Address = I1	Segments Left = 3
	Address[1] = I2
	Address[2] = I3
	Address[3] = D

As the packet travels from I1 to I2:

Source Address = S	Hdr Ext Len = 6
Destination Address = I2	Segments Left = 2
	Address[1] = I1
	Address[2] = I3
	Address[3] = D

As the packet travels from I2 to I3:

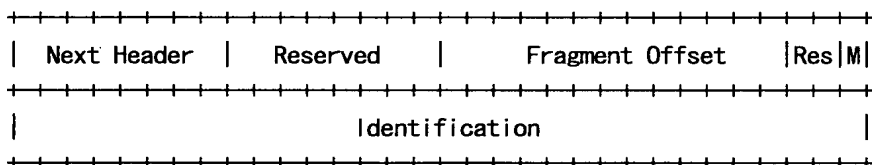
Source Address = S	Hdr Ext Len = 6
Destination Address = I3	Segments Left = 1
	Address[1] = I1
	Address[2] = I2
	Address[3] = D

As the packet travels from I3 to D:

Source Address = S	Hdr Ext Len = 6
Destination Address = D	Segments Left = 0
	Address[1] = I1
	Address[2] = I2
	Address[3] = I3

4.5 Fragment Header

The Fragment header is used by an IPv6 source to send a packet larger than would fit in the path MTU to its destination. (Note: unlike IPv4, fragmentation in IPv6 is performed only by source nodes, not by routers along a packet's delivery path — see section 5.) The Fragment header is identified by a Next Header value of 44 in the immediately preceding header, and has the following format:



Next Header	8-bit selector. Identifies the initial header type of the Fragmentable Part of the original packet (defined below). Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
Reserved	8-bit reserved field. Initialized to zero for transmission; ignored on reception.
Fragment Offset	13-bit unsigned integer. The offset, in 8-octet units, of the data following this header, relative to the start of the Fragmentable Part of the original packet.
Res	2-bit reserved field. Initialized to zero for transmission; ignored on reception.
M flag	1 = more fragments; 0 = last fragment.
Identification	32 bits. See description below.

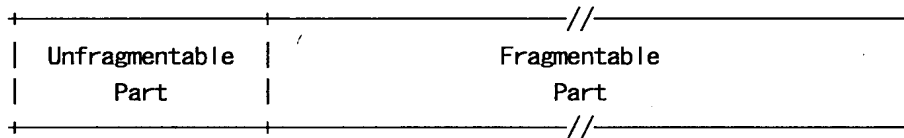
In order to send a packet that is too large to fit in the MTU of the path to its destination, a source node may divide the packet into fragments and send each fragment as a separate packet, to be reassembled at the receiver.

For every packet that is to be fragmented, the source node generates an Identification value. The Identification must be different than that of any other fragmented packet sent recently* with the same Source Address and Destination Address. If a Routing header is present, the Destination Address of concern is that of the final destination.

* "recently" means within the maximum likely lifetime of a packet, including transit time from source to destination and time spent awaiting reassembly with other fragments of the same packet. However, it is not required that a source node know the maximum packet lifetime. Rather, it is assumed that the requirement can be met by maintaining the Identification value as a simple, 32-bit, "wrap-around" counter, incremented each time a packet must be fragmented. It is an implementation choice whether to maintain a single counter for the node or multiple counters, e.g., one for each of the node's possible source addresses, or one for each active (source address, destination address) combination.

The initial, large, unfragmented packet is referred to as the "original packet", and it is considered to consist of two parts, as illustrated:

original packet:

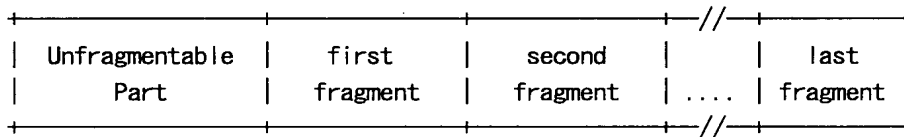


The Unfragmentable Part consists of the IPv6 header plus any extension headers that must be processed by nodes en route to the destination, that is, all headers up to and including the Routing header if present, else the Hop-by-Hop Options header if present, else no extension headers.

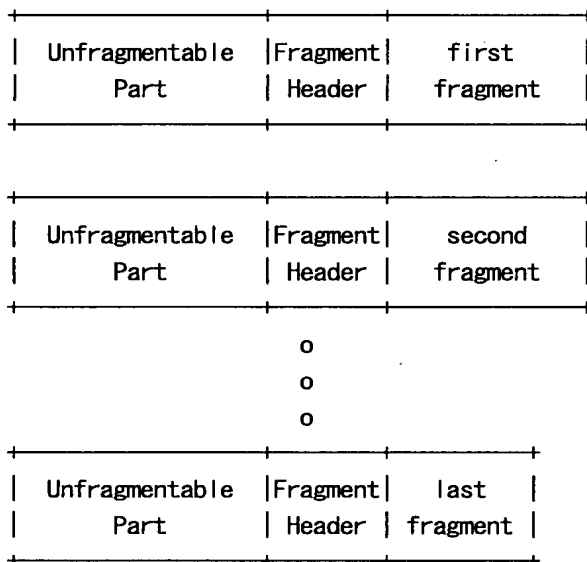
The Fragmentable Part consists of the rest of the packet, that is, any extension headers that need be processed only by the final destination node(s), plus the upper-layer header and data.

The Fragmentable Part of the original packet is divided into fragments, each, except possibly the last ("rightmost") one, being an integer multiple of 8 octets long. The fragments are transmitted in separate "fragment packets" as illustrated:

original packet:



fragment packets:



Each fragment packet is composed of:

- (1) The Unfragmentable Part of the original packet, with the Payload Length of the original IPv6 header changed to contain the length of this fragment packet only (excluding the length of the IPv6 header itself), and the Next Header field of the last header of the Unfragmentable Part changed to 44.
- (2) A Fragment header containing:

The Next Header value that identifies the first header of the Fragmentable Part of the original packet.

A Fragment Offset containing the offset of the fragment, in 8-octet units, relative to the start of the Fragmentable Part of the original packet. The Fragment Offset of the first ("leftmost") fragment is 0.

An M flag value of 0 if the fragment is the last ("rightmost") one, else an M flag value of 1.

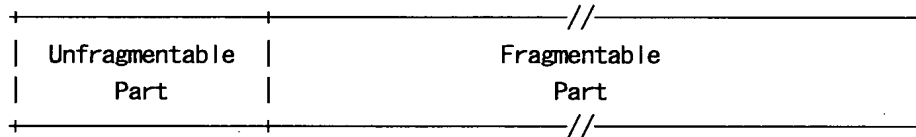
The Identification value generated for the original packet.

- (3) The fragment itself.

The lengths of the fragments must be chosen such that the resulting fragment packets fit within the MTU of the path to the packets' destination(s).

At the destination, fragment packets are reassembled into their original, unfragmented form, as illustrated:

reassembled original packet:



The following rules govern reassembly:

An original packet is reassembled only from fragment packets that have the same Source Address, Destination Address, and Fragment Identification.

The Unfragmentable Part of the reassembled packet consists of all headers up to, but not including, the Fragment header of the first fragment packet (that is, the packet whose Fragment Offset is zero), with the following two changes:

The Next Header field of the last header of the Unfragmentable Part is obtained from the Next Header field of the first fragment's Fragment header.

The Payload Length of the reassembled packet is computed from the length of the Unfragmentable Part and the length and offset of the last fragment. For example, a formula for computing the Payload Length of the reassembled original packet is:

$$PL.orig = PL.first - FL.first - 8 + (8 * FO.last) + FL.last$$

where

PL.orig = Payload Length field of reassembled packet.

PL.first = Payload Length field of first fragment packet.

FL.first = length of fragment following Fragment header of first fragment packet.

FO.last = Fragment Offset field of Fragment header of last fragment packet.

FL.last = length of fragment following Fragment header of last fragment packet.

The Fragmentable Part of the reassembled packet is constructed from the fragments following the Fragment headers in each of the fragment packets. The length of each fragment is computed by subtracting from the packet's Payload Length the length of the

headers between the IPv6 header and fragment itself; its relative position in Fragmentable Part is computed from its Fragment Offset value.

The Fragment header is not present in the final, reassembled packet.

The following error conditions may arise when reassembling fragmented packets:

If insufficient fragments are received to complete reassembly of a packet within 60 seconds of the reception of the first-arriving fragment of that packet, reassembly of that packet must be abandoned and all the fragments that have been received for that packet must be discarded. If the first fragment (i.e., the one with a Fragment Offset of zero) has been received, an ICMP Time Exceeded — Fragment Reassembly Time Exceeded message should be sent to the source of that fragment.

If the length of a fragment, as derived from the fragment packet's Payload Length field, is not a multiple of 8 octets and the M flag of that fragment is 1, then that fragment must be discarded and an ICMP Parameter Problem, Code 0, message should be sent to the source of the fragment, pointing to the Payload Length field of the fragment packet.

If the length and offset of a fragment are such that the Payload Length of the packet reassembled from that fragment would exceed 65,535 octets, then that fragment must be discarded and an ICMP Parameter Problem, Code 0, message should be sent to the source of the fragment, pointing to the Fragment Offset field of the fragment packet.

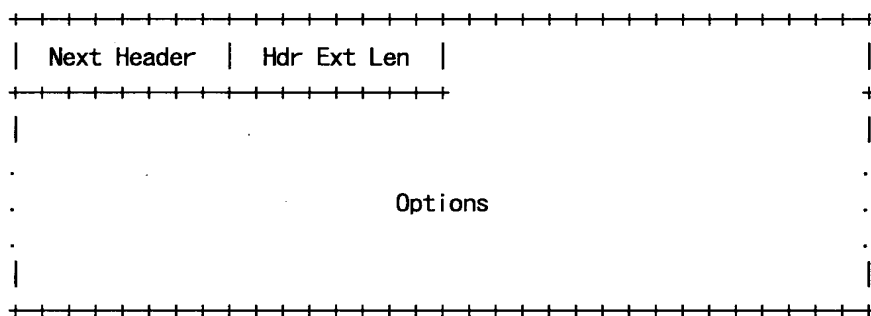
The following conditions are not expected to occur, but are not considered errors if they do:

The number and content of the headers preceding the Fragment header of different fragments of the same original packet may differ. Whatever headers are present, preceding the Fragment header in each fragment packet, are processed when the packets arrive, prior to queueing the fragments for reassembly. Only those headers in the Offset zero fragment packet are retained in the reassembled packet.

The Next Header values in the Fragment headers of different fragments of the same original packet may differ. Only the value from the Offset zero fragment packet is used for reassembly.

4.6 Destination Options Header

The Destination Options header is used to carry optional information that need be examined only by a packet's destination node(s). The Destination Options header is identified by a Next Header value of 60 in the immediately preceding header, and has the following format:



Next Header	8-bit selector. Identifies the type of header immediately following the Destination Options header. Uses the same values as the IPv4 Protocol field [RFC-1700 et seq.].
Hdr Ext Len	8-bit unsigned integer. Length of the Destination Options header in 8-octet units, not including the first 8 octets.
Options	Variable-length field, of length such that the complete Destination Options header is an integer multiple of 8 octets long. Contains one or more TLV-encoded options, as described in section 4.2.

The only destination options defined in this document are the Pad1 and PadN options specified in section 4.2.

Note that there are two possible ways to encode optional destination information in an IPv6 packet: either as an option in the Destination Options header, or as a separate extension header. The Fragment header and the Authentication header are examples of the latter approach. Which approach can be used depends on what action is desired of a destination node that does not understand the optional information:

- o If the desired action is for the destination node to discard the packet and, only if the packet's Destination Address is not a multicast address, send an ICMP Unrecognized Type message to the packet's Source Address, then the information may be encoded either as a separate header or as an option in the

Destination Options header whose Option Type has the value 11 in its highest-order two bits. The choice may depend on such factors as which takes fewer octets, or which yields better alignment or more efficient parsing.

- o If any other action is desired, the information must be encoded as an option in the Destination Options header whose Option Type has the value 00, 01, or 10 in its highest-order two bits, specifying the desired action (see section 4.2).

4.7 No Next Header

The value 59 in the Next Header field of an IPv6 header or any extension header indicates that there is nothing following that header. If the Payload Length field of the IPv6 header indicates the presence of octets past the end of a header whose Next Header field contains 59, those octets must be ignored, and passed on unchanged if the packet is forwarded.

5. Packet Size Issues

IPv6 requires that every link in the internet have an MTU of 1280 octets or greater. On any link that cannot convey a 1280-octet packet in one piece, link-specific fragmentation and reassembly must be provided at a layer below IPv6.

Links that have a configurable MTU (for example, PPP links [RFC-1661]) must be configured to have an MTU of at least 1280 octets; it is recommended that they be configured with an MTU of 1500 octets or greater, to accommodate possible encapsulations (i.e., tunneling) without incurring IPv6-layer fragmentation.

From each link to which a node is directly attached, the node must be able to accept packets as large as that link's MTU.

It is strongly recommended that IPv6 nodes implement Path MTU Discovery [RFC-1981], in order to discover and take advantage of path MTUs greater than 1280 octets. However, a minimal IPv6 implementation (e.g., in a boot ROM) may simply restrict itself to sending packets no larger than 1280 octets, and omit implementation of Path MTU Discovery.

In order to send a packet larger than a path's MTU, a node may use the IPv6 Fragment header to fragment the packet at the source and have it reassembled at the destination(s). However, the use of such fragmentation is discouraged in any application that is able to adjust its packets to fit the measured path MTU (i.e., down to 1280 octets).

A node must be able to accept a fragmented packet that, after reassembly, is as large as 1500 octets. A node is permitted to accept fragmented packets that reassemble to more than 1500 octets. An upper-layer protocol or application that depends on IPv6 fragmentation to send packets larger than the MTU of a path should not send packets larger than 1500 octets unless it has assurance that the destination is capable of reassembling packets of that larger size.

In response to an IPv6 packet that is sent to an IPv4 destination (i.e., a packet that undergoes translation from IPv6 to IPv4), the originating IPv6 node may receive an ICMP Packet Too Big message reporting a Next-Hop MTU less than 1280. In that case, the IPv6 node is not required to reduce the size of subsequent packets to less than 1280, but must include a Fragment header in those packets so that the IPv6-to-IPv4 translating router can obtain a suitable Identification value to use in resulting IPv4 fragments. Note that this means the payload may have to be reduced to 1232 octets (1280 minus 40 for the IPv6 header and 8 for the Fragment header), and smaller still if additional extension headers are used.

6. Flow Labels

The 20-bit Flow Label field in the IPv6 header may be used by a source to label sequences of packets for which it requests special handling by the IPv6 routers, such as non-default quality of service or "real-time" service. This aspect of IPv6 is, at the time of writing, still experimental and subject to change as the requirements for flow support in the Internet become clearer. Hosts or routers that do not support the functions of the Flow Label field are required to set the field to zero when originating a packet, pass the field on unchanged when forwarding a packet, and ignore the field when receiving a packet.

Appendix A describes the current intended semantics and usage of the Flow Label field.

7. Traffic Classes

The 8-bit Traffic Class field in the IPv6 header is available for use by originating nodes and/or forwarding routers to identify and distinguish between different classes or priorities of IPv6 packets. At the point in time at which this specification is being written, there are a number of experiments underway in the use of the IPv4 Type of Service and/or Precedence bits to provide various forms of "differentiated service" for IP packets, other than through the use of explicit flow set-up. The Traffic Class field in the IPv6 header is intended to allow similar functionality to be supported in IPv6.

It is hoped that those experiments will eventually lead to agreement on what sorts of traffic classifications are most useful for IP packets. Detailed definitions of the syntax and semantics of all or some of the IPv6 Traffic Class bits, whether experimental or intended for eventual standardization, are to be provided in separate documents.

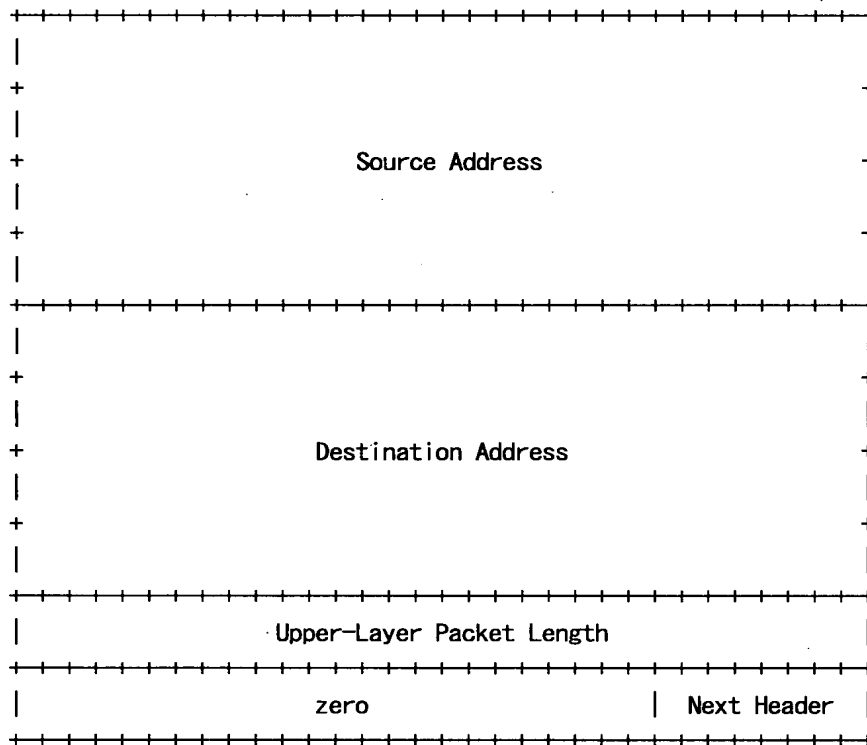
The following general requirements apply to the Traffic Class field:

- o The service interface to the IPv6 service within a node must provide a means for an upper-layer protocol to supply the value of the Traffic Class bits in packets originated by that upper-layer protocol. The default value must be zero for all 8 bits.
- o Nodes that support a specific (experimental or eventual standard) use of some or all of the Traffic Class bits are permitted to change the value of those bits in packets that they originate, forward, or receive, as required for that specific use. Nodes should ignore and leave unchanged any bits of the Traffic Class field for which they do not support a specific use.
- o An upper-layer protocol must not assume that the value of the Traffic Class bits in a received packet are the same as the value sent by the packet's source.

8. Upper-Layer Protocol Issues

8.1 Upper-Layer Checksums

Any transport or other upper-layer protocol that includes the addresses from the IP header in its checksum computation must be modified for use over IPv6, to include the 128-bit IPv6 addresses instead of 32-bit IPv4 addresses. In particular, the following illustration shows the TCP and UDP "pseudo-header" for IPv6:



- o If the IPv6 packet contains a Routing header, the Destination Address used in the pseudo-header is that of the final destination. At the originating node, that address will be in the last element of the Routing header; at the recipient(s), that address will be in the Destination Address field of the IPv6 header.
- o The Next Header value in the pseudo-header identifies the upper-layer protocol (e.g., 6 for TCP, or 17 for UDP). It will differ from the Next Header value in the IPv6 header if there are extension headers between the IPv6 header and the upper-layer header.
- o The Upper-Layer Packet Length in the pseudo-header is the length of the upper-layer header and data (e.g., TCP header plus TCP data). Some upper-layer protocols carry their own

length information (e.g., the Length field in the UDP header); for such protocols, that is the length used in the pseudo-header. Other protocols (such as TCP) do not carry their own length information, in which case the length used in the pseudo-header is the Payload Length from the IPv6 header, minus the length of any extension headers present between the IPv6 header and the upper-layer header.

- o Unlike IPv4, when UDP packets are originated by an IPv6 node, the UDP checksum is not optional. That is, whenever originating a UDP packet, an IPv6 node must compute a UDP checksum over the packet and the pseudo-header, and, if that computation yields a result of zero, it must be changed to hex FFFF for placement in the UDP header. IPv6 receivers must discard UDP packets containing a zero checksum, and should log the error.

The IPv6 version of ICMP [ICMPv6] includes the above pseudo-header in its checksum computation; this is a change from the IPv4 version of ICMP, which does not include a pseudo-header in its checksum. The reason for the change is to protect ICMP from misdelivery or corruption of those fields of the IPv6 header on which it depends, which, unlike IPv4, are not covered by an internet-layer checksum. The Next Header field in the pseudo-header for ICMP contains the value 58, which identifies the IPv6 version of ICMP.

8.2 Maximum Packet Lifetime

Unlike IPv4, IPv6 nodes are not required to enforce maximum packet lifetime. That is the reason the IPv4 "Time to Live" field was renamed "Hop Limit" in IPv6. In practice, very few, if any, IPv4 implementations conform to the requirement that they limit packet lifetime, so this is not a change in practice. Any upper-layer protocol that relies on the internet layer (whether IPv4 or IPv6) to limit packet lifetime ought to be upgraded to provide its own mechanisms for detecting and discarding obsolete packets.

8.3 Maximum Upper-Layer Payload Size

When computing the maximum payload size available for upper-layer data, an upper-layer protocol must take into account the larger size of the IPv6 header relative to the IPv4 header. For example, in IPv4, TCP's MSS option is computed as the maximum packet size (a default value or a value learned through Path MTU Discovery) minus 40 octets (20 octets for the minimum-length IPv4 header and 20 octets for the minimum-length TCP header). When using TCP over IPv6, the MSS must be computed as the maximum packet size minus 60 octets,

because the minimum-length IPv6 header (i.e., an IPv6 header with no extension headers) is 20 octets longer than a minimum-length IPv4 header.

8.4 Responding to Packets Carrying Routing Headers

When an upper-layer protocol sends one or more packets in response to a received packet that included a Routing header, the response packet(s) must not include a Routing header that was automatically derived by "reversing" the received Routing header UNLESS the integrity and authenticity of the received Source Address and Routing header have been verified (e.g., via the use of an Authentication header in the received packet). In other words, only the following kinds of packets are permitted in response to a received packet bearing a Routing header:

- o Response packets that do not carry Routing headers.
- o Response packets that carry Routing headers that were NOT derived by reversing the Routing header of the received packet (for example, a Routing header supplied by local configuration).
- o Response packets that carry Routing headers that were derived by reversing the Routing header of the received packet IF AND ONLY IF the integrity and authenticity of the Source Address and Routing header from the received packet have been verified by the responder.

Appendix A. Semantics and Usage of the Flow Label Field

A flow is a sequence of packets sent from a particular source to a particular (unicast or multicast) destination for which the source desires special handling by the intervening routers. The nature of that special handling might be conveyed to the routers by a control protocol, such as a resource reservation protocol, or by information within the flow's packets themselves, e.g., in a hop-by-hop option. The details of such control protocols or options are beyond the scope of this document.

There may be multiple active flows from a source to a destination, as well as traffic that is not associated with any flow. A flow is uniquely identified by the combination of a source address and a non-zero flow label. Packets that do not belong to a flow carry a flow label of zero.

A flow label is assigned to a flow by the flow's source node. New flow labels must be chosen (pseudo-)randomly and uniformly from the range 1 to FFFFF hex. The purpose of the random allocation is to make any set of bits within the Flow Label field suitable for use as a hash key by routers, for looking up the state associated with the flow.

All packets belonging to the same flow must be sent with the same source address, destination address, and flow label. If any of those packets includes a Hop-by-Hop Options header, then they all must be originated with the same Hop-by-Hop Options header contents (excluding the Next Header field of the Hop-by-Hop Options header). If any of those packets includes a Routing header, then they all must be originated with the same contents in all extension headers up to and including the Routing header (excluding the Next Header field in the Routing header). The routers or destinations are permitted, but not required, to verify that these conditions are satisfied. If a violation is detected, it should be reported to the source by an ICMP Parameter Problem message, Code 0, pointing to the high-order octet of the Flow Label field (i.e., offset 1 within the IPv6 packet).

The maximum lifetime of any flow-handling state established along a flow's path must be specified as part of the description of the state-establishment mechanism, e.g., the resource reservation protocol or the flow-setup hop-by-hop option. A source must not re-use a flow label for a new flow within the maximum lifetime of any flow-handling state that might have been established for the prior use of that flow label.

When a node stops and restarts (e.g., as a result of a "crash"), it must be careful not to use a flow label that it might have used for an earlier flow whose lifetime may not have expired yet. This may be accomplished by recording flow label usage on stable storage so that it can be remembered across crashes, or by refraining from using any flow labels until the maximum lifetime of any possible previously established flows has expired. If the minimum time for rebooting the node is known, that time can be deducted from the necessary waiting period before starting to allocate flow labels.

There is no requirement that all, or even most, packets belong to flows, i.e., carry non-zero flow labels. This observation is placed here to remind protocol designers and implementors not to assume otherwise. For example, it would be unwise to design a router whose performance would be adequate only if most packets belonged to flows, or to design a header compression scheme that only worked on packets that belonged to flows.

Appendix B. Formatting Guidelines for Options

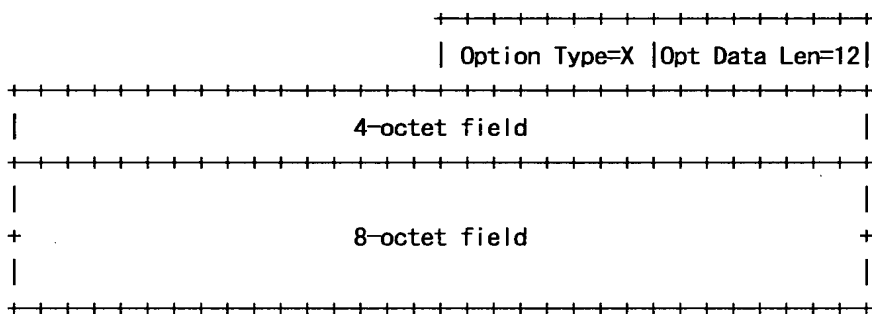
This appendix gives some advice on how to lay out the fields when designing new options to be used in the Hop-by-Hop Options header or the Destination Options header, as described in section 4.2. These guidelines are based on the following assumptions:

- o One desirable feature is that any multi-octet fields within the Option Data area of an option be aligned on their natural boundaries, i.e., fields of width n octets should be placed at an integer multiple of n octets from the start of the Hop-by-Hop or Destination Options header, for $n = 1, 2, 4$, or 8 .
- o Another desirable feature is that the Hop-by-Hop or Destination Options header take up as little space as possible, subject to the requirement that the header be an integer multiple of 8 octets long.
- o It may be assumed that, when either of the option-bearing headers are present, they carry a very small number of options, usually only one.

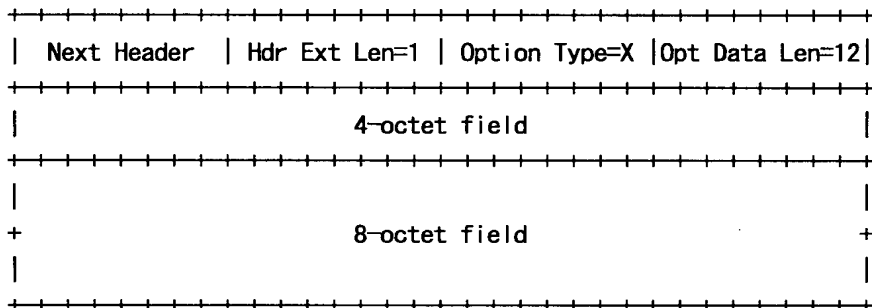
These assumptions suggest the following approach to laying out the fields of an option: order the fields from smallest to largest, with no interior padding, then derive the alignment requirement for the entire option based on the alignment requirement of the largest field (up to a maximum alignment of 8 octets). This approach is illustrated in the following examples:

Example 1

If an option X required two data fields, one of length 8 octets and one of length 4 octets, it would be laid out as follows:

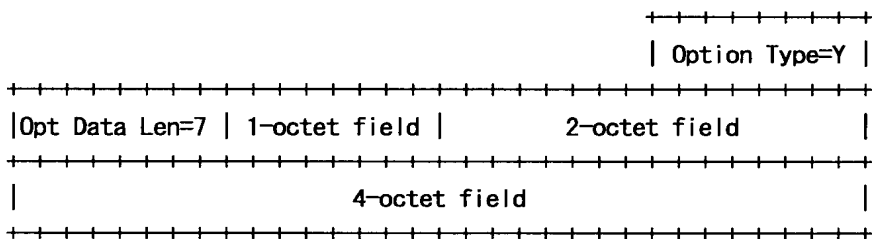


Its alignment requirement is $8n+2$, to ensure that the 8-octet field starts at a multiple-of-8 offset from the start of the enclosing header. A complete Hop-by-Hop or Destination Options header containing this one option would look as follows:

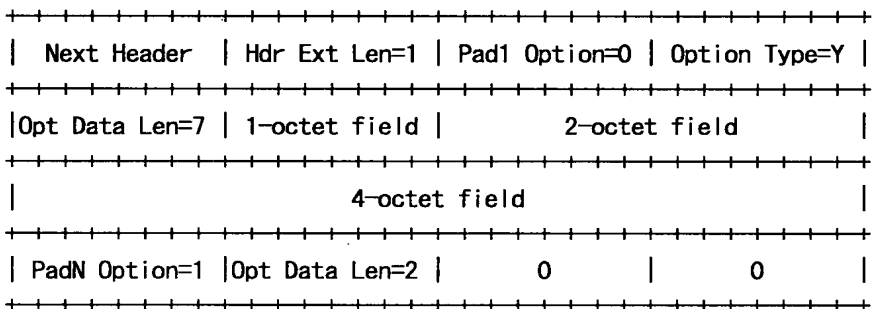


Example 2

If an option Y required three data fields, one of length 4 octets, one of length 2 octets, and one of length 1 octet, it would be laid out as follows:

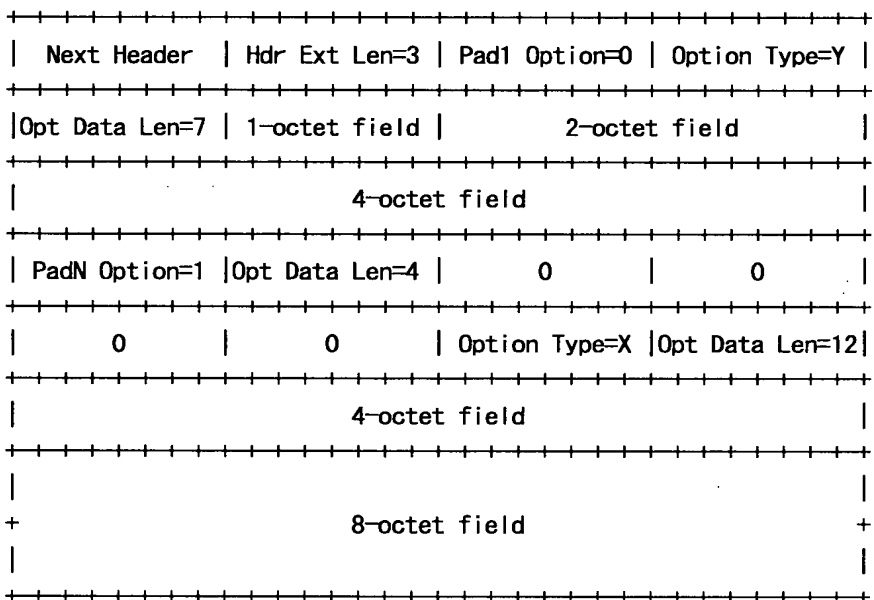
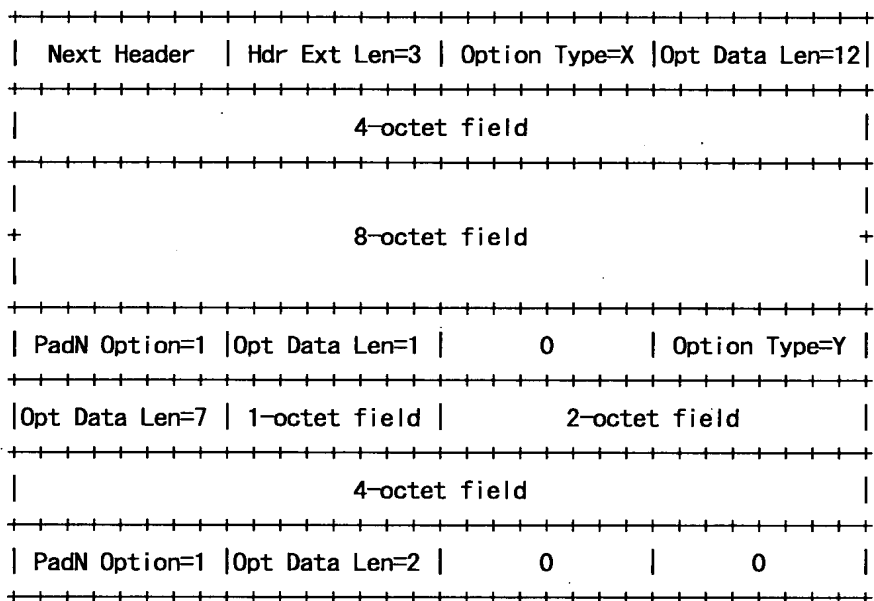


Its alignment requirement is $4n+3$, to ensure that the 4-octet field starts at a multiple-of-4 offset from the start of the enclosing header. A complete Hop-by-Hop or Destination Options header containing this one option would look as follows:



Example 3

A Hop-by-Hop or Destination Options header containing both options X and Y from Examples 1 and 2 would have one of the two following formats, depending on which option appeared first:



Security Considerations

The security features of IPv6 are described in the Security Architecture for the Internet Protocol [RFC-2401].

Acknowledgments

The authors gratefully acknowledge the many helpful suggestions of the members of the IPng working group, the End-to-End Protocols research group, and the Internet Community At Large.

Authors' Addresses

Stephen E. Deering
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA

Phone: +1 408 527 8213
Fax: +1 408 527 8254
EMail: deering@cisco.com

Robert M. Hinden
Nokia
232 Java Drive
Sunnyvale, CA 94089
USA

Phone: +1 408 990-2004
Fax: +1 408 743-5677
EMail: hinden@iprg.nokia.com

References

- [RFC-2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [RFC-2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.
- [RFC-2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Protocol (ESP)", RFC 2406, November 1998.
- [ICMPv6] Conta, A. and S. Deering, "ICMP for the Internet Protocol Version 6 (IPv6)", RFC 2463, December 1998.

- [ADDRARCH] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.
- [RFC-1981] McCann, J., Mogul, J. and S. Deering, "Path MTU Discovery for IP version 6", RFC 1981, August 1996.
- [RFC-791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.
- [RFC-1700] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994. See also:
<http://www.iana.org/numbers.html>
- [RFC-1661] Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.

CHANGES SINCE RFC-1883

This memo has the following changes from RFC-1883. Numbers identify the Internet-Draft version in which the change was made.

- 02) Removed all references to jumbograms and the Jumbo Payload option (moved to a separate document).
- 02) Moved most of Flow Label description from section 6 to (new) Appendix A.
- 02) In Flow Label description, now in Appendix A, corrected maximum Flow Label value from FFFFFFFF to FFFFFF (i.e., one less "F") due to reduction of size of Flow Label field from 24 bits to 20 bits.
- 02) Renumbered (relettered?) the previous Appendix A to be Appendix B.
- 02) Changed the wording of the Security Considerations section to avoid dependency loop between this spec and the IPsec specs.
- 02) Updated R. Hinden's email address and company affiliation.

-
- 01) In section 3, changed field name "Class" to "Traffic Class" and increased its size from 4 to 8 bits. Decreased size of Flow Label field from 24 to 20 bits to compensate for increase in Traffic Class field.

- 01) In section 4.1, restored the order of the Authentication Header and the ESP header, which were mistakenly swapped in the 00 version of this memo.
 - 01) In section 4.4, deleted the Strict/Loose Bit Map field and the strict routing functionality from the Type 0 Routing header, and removed the restriction on number of addresses that may be carried in the Type 0 Routing header (was limited to 23 addresses, because of the size of the strict/loose bit map).
 - 01) In section 5, changed the minimum IPv6 MTU from 576 to 1280 octets, and added a recommendation that links with configurable MTU (e.g., PPP links) be configured to have an MTU of at least 1500 octets.
 - 01) In section 5, deleted the requirement that a node must not send fragmented packets that reassemble to more than 1500 octets without knowledge of the destination reassembly buffer size, and replaced it with a recommendation that upper-layer protocols or applications should not do that.
 - 01) Replaced reference to the IPv4 Path MTU Discovery spec (RFC-1191) with reference to the IPv6 Path MTU Discovery spec (RFC-1981), and deleted the Notes at the end of section 5 regarding Path MTU Discovery, since those details are now covered by RFC-1981.
 - 01) In section 6, deleted specification of "opportunistic" flow set-up, and removed all references to the 6-second maximum lifetime for opportunistically established flow state.
 - 01) In section 7, deleted the provisional description of the internal structure and semantics of the Traffic Class field, and specified that such descriptions be provided in separate documents.
-
- 00) In section 4, corrected the Code value to indicate "unrecognized Next Header type encountered" in an ICMP Parameter Problem message (changed from 2 to 1).
 - 00) In the description of the Payload Length field in section 3, and of the Jumbo Payload Length field in section 4.3, made it clearer that extension headers are included in the payload length count.

- 00) In section 4.1, swapped the order of the Authentication header and the ESP header. (NOTE: this was a mistake, and the change was undone in version 01.)
 - 00) In section 4.2, made it clearer that options are identified by the full 8-bit Option Type, not by the low-order 5 bits of an Option Type. Also specified that the same Option Type numbering space is used for both Hop-by-Hop Options and Destination Options headers.
 - 00) In section 4.4, added a sentence requiring that nodes processing a Routing header must send an ICMP Packet Too Big message in response to a packet that is too big to fit in the next hop link (rather than, say, performing fragmentation).
 - 00) Changed the name of the IPv6 Priority field to "Class", and replaced the previous description of Priority in section 7 with a description of the Class field. Also, excluded this field from the set of fields that must remain the same for all packets in the same flow, as specified in section 6.
 - 00) In the pseudo-header in section 8.1, changed the name of the "Payload Length" field to "Upper-Layer Packet Length". Also clarified that, in the case of protocols that carry their own length info (like non-jumbogram UDP), it is the upper-layer-derived length, not the IP-layer-derived length, that is used in the pseudo-header.
 - 00) Added section 8.4, specifying that upper-layer protocols, when responding to a received packet that carried a Routing header, must not include the reverse of the Routing header in the response packet(s) unless the received Routing header was authenticated.
 - 00) Fixed some typos and grammatical errors.
 - 00) Authors' contact info updated.
-

Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IP in IP Tunneling

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard. Distribution of this memo is unlimited.

IESG Note:

Note that this memo is an individual effort of the author. This document reflects a current informal practice in the internet. There is an effort underway within the IETF Mobile-IP Working Group to provide an appropriate proposed standard to address this issue.

Abstract

This document discusses implementation techniques for using IP Protocol/Payload number 4 Encapsulation for tunneling with IP Security and other protocols.

Table of Contents

1.	Introduction	2
2.	Encapsulation	3
3.	Tunnel Management	5
3.1	Tunnel MTU Discovery	5
3.2	Congestion	6
3.3	Routing Failures	6
3.4	Other ICMP Messages	6
	SECURITY CONSIDERATIONS	7
	REFERENCES	7
	ACKNOWLEDGEMENTS	8
	AUTHOR'S ADDRESS	8

1. Introduction

The IP in IP encapsulation Protocol/Payload number 4 [RFC-1700] has long been used to bridge portions of the Internet which have disjoint capabilities or policies. This document describes implementation techniques used for many years by the Amateur Packet Radio network for joining a large mobile network, and also by early implementations of IP Security protocols.

Use of IP in IP encapsulation differs from later tunneling techniques (for example, protocol numbers 98 [RFC-1241], 94 [IDM91a], 53 [swIPe], and 47 [RFC-1701]) in that it does not insert its own special glue header between IP headers. Instead, the original unadorned IP Header is retained, and simply wrapped in another standard IP header.

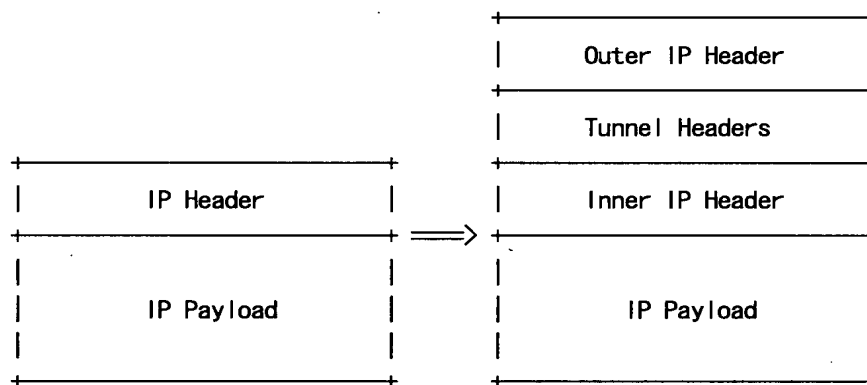
This information applies principally to encapsulation of IP version 4. Other IP versions will be described in separate documents.

2. Encapsulation

The encapsulation technique is fairly simple. An outer IP header is added before the original IP header. Between them are any other headers for the path, such as security headers specific to the tunnel configuration.

The outer IP header Source and Destination identify the "endpoints" of the tunnel. The inner IP header Source and Destination identify the original sender and recipient of the datagram.

Each header chains to the next using IP Protocol values [RFC-1700].



The format of IP headers is described in [RFC-791].

Type Of Service copied from the inner IP header. Optionally, another TOS may be used between cooperating peers.

This is in keeping with the transparency principle that if the user was expecting a given level of service, then the tunnel should provide the same service. However, some tunnels may be constructed specifically to provide a different level of service as a matter of policy.

Identification A new number is generated for each outer IP header.

The encapsulated datagram may have already been fragmented, and another level of fragmentation may occur due to the tunnel encapsulation. These tunnel fragments will be reassembled by the decapsulator, rather than the final destination.

Reserved

ignored (set to zero).

This unofficial flag has seen experimental use, and while it remains in the inner IP header, does not affect the tunnel.

Don't Fragment copied from the inner IP header. This allows the originator to control the level of performance tradeoffs. See "Tunnel MTU Discovery".

More Fragments set as required when fragmenting.

The flag is not copied for the same reason that a separate Identification is used.

Time To Live the default value specified in the most recent "Assigned Numbers" [RFC-1700]. This ensures that long unanticipated tunnels do not interrupt the flow of datagrams between endpoints.

The inner TTL is decremented once before encapsulation, and is not affected by decapsulation.

Protocol the next header: 4 for the inner IP header, when no intervening tunnel headers are in use.

Source an IP address associated with the interface used to send the datagram.

Destination an IP address of the tunnel decapsulator.

Options not copied from the inner IP header. However, new options particular to the path MAY be added.

Timestamp, Loose Source Route, Strict Source Route, and Record Route are deliberately hidden within the tunnel. Often, tunnels are constructed to overcome the inadequacies of these options.

Any supported flavors of security options of the inner IP header MAY affect the choice of security options for the tunnel. It is not expected that there be a one-to-one mapping of such options to the options or security headers selected for the tunnel.

3. Tunnel Management

It is possible that one of the routers along the tunnel interior might encounter an error while processing the datagram, causing it to return an ICMP [RFC-792] error message to the encapsulator at the IP Source of the tunnel. Unfortunately, ICMP only requires IP routers to return 8 bytes (64 bits) of the datagram beyond the IP header. This is not enough to include the entire encapsulated header. Thus, it is not generally possible for an encapsulating router to immediately reflect an ICMP message from the interior of a tunnel back to the originating host.

However, by carefully maintaining "soft state" about its tunnels, the encapsulator can return accurate ICMP messages in most cases. The router **SHOULD** maintain at least the following soft state information about each tunnel:

- Reachability of the end of the tunnel.
- Congestion of the tunnel.
- MTU of the tunnel.

The router uses the ICMP messages it receives from the interior of a tunnel to update the soft state information for that tunnel. When subsequent datagrams arrive that would transit the tunnel, the router checks the soft state for the tunnel. If the datagram would violate the state of the tunnel (such as the MTU is greater than the tunnel MTU when Don't Fragment is set), the router sends an appropriate ICMP error message back to the originator, but also forwards the datagram into the tunnel. Forwarding the datagram despite returning the error message ensures that changes in tunnel state will be learned.

Using this technique, the ICMP error messages from encapsulating routers will not always match one-to-one with errors encountered within the tunnel, but they will accurately reflect the state of the network.

3.1. Tunnel MTU Discovery

When the Don't Fragment bit is set by the originator and copied into the outer IP header, the proper MTU of the tunnel will be learned from ICMP (Type 3 Code 4) "Datagram Too Big" errors reported to the encapsulator. To support originating hosts which use this capability, all implementations **MUST** support Path MTU Discovery [RFC-1191, RFC-1435] within their tunnels.

As a benefit of Tunnel MTU Discovery, any fragmentation which occurs because of the size of the encapsulation header is done only once after encapsulation. This prevents more than one fragmentation of a single datagram, which improves processing efficiency of the path routers and tunnel decapsulator.

3.2. Congestion

Tunnel soft state will collect indications of congestion, such as an ICMP (Type 4) Source Quench in datagrams from the decapsulator (tunnel peer). When forwarding another datagram into the tunnel, it is appropriate to send Source Quench messages to the originator.

3.3. Routing Failures

Because the TTL is reset each time that a datagram is encapsulated, routing loops within a tunnel are particularly dangerous when they arrive again at the encapsulator. If the IP Source matches any of its interfaces, an implementation **MUST NOT** further encapsulate. Instead, the datagram is forwarded normally.

ICMP (Type 11) Time Exceeded messages report routing loops within the tunnel itself. ICMP (Type 3) Destination Unreachable messages report delivery failures to the decapsulator. This soft state **MUST** be reported to the originator as (Type 3 Code 0) Network Unreachable.

3.4. Other ICMP Messages

Most ICMP error messages are not relevant to the use of the tunnel. In particular, parameter problems are likely to be a result of misconfiguration of the encapsulator, and **MUST NOT** be reported to the originator.

Security Considerations

Security issues are briefly discussed in this memo. The use of tunneling may obviate some older IP security options (labelling), but will better support newer IP Security headers.

References

- [IDM91a] Ioannidis, J., Duchamp, D., Maguire, G., "IP-based protocols for mobile internetworking", Proceedings of SIGCOMM '91, ACM, September 1991.
- [RFC-791]
Postel, J., "Internet Protocol", STD 5, RFC 791, USC/Information Sciences Institute, September 1981.
- [RFC-792]
Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, USC/Information Sciences Institute, September 1981.
- [RFC-1191]
Mogul, J., and S. Deering, "Path MTU Discovery", RFC 1191, DECWRL, Stanford University, November 1990.
- [RFC-1241]
Mills, D., and R. Woodburn, "A Scheme for an Internet Encapsulation Protocol: Version 1", UDEL, July 1991.
- [RFC-1435]
Knowles, S., "IESG Advice from Experience with Path MTU Discovery", RFC 1435, FTP Software, March 1993.
- [RFC-1700]
Reynolds, J., and J. Postel, "Assigned Numbers", STD 2, RFC 1700, USC/Information Sciences Institute, October 1994.
- [RFC-1701]
Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation (GRE)", RFC 1701, October 1994.
- [swlPe] Ioannidis, J., and Blaze, M., "The Architecture and Implementation of Network-Layer Security Under Unix", Fourth Usenix Security Symposium Proceedings, October 1993.

Acknowledgements

These implementation details of IP Tunneling are derived in large part from independent work in 1990 by Phil Karn and the TCP-Group hams using KA9Q NOS.

Special thanks to John Ioannidis (then of Columbia University) for inspiration and experimentation which began this most recent round of IP Mobility and IP Security development. Some of this text was derived from [IDM91a] and [swIPe].

The chaining of headers was also described in "Simple Internet Protocol", by Steve Deering (Xerox PARC).

The overall organization and some of this text was derived from [RFC-1241], by David Mills (U Delaware) and Robert Woodburn (SAIC).

Some of the text on tunnel soft state was derived from "IP Address Encapsulation (IPAE)", by Robert E. Gilligan, Erik Nordmark, and Bob Hinden (all of Sun Microsystems).

Author's Address

Questions about this memo can also be directed to:

William Allen Simpson
Daydreamer
Computer Systems Consulting Services
1384 Fontaine
Madison Heights, Michigan 48071

Bill.Simpson@um.cc.umich.edu
bsimpson@MorningStar.com

Generic Packet Tunneling in IPv6 Specification

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This document defines the model and generic mechanisms for IPv6 encapsulation of Internet packets, such as IPv6 and IPv4. The model and mechanisms can be applied to other protocol packets as well, such as AppleTalk, IPX, CLNP, or others.

Table of Contents

1. Introduction.....	2
2. Terminology.....	2
3. IPv6 Tunneling.....	4
3.1 IPv6 Encapsulation.....	6
3.2 IPv6 Packet Processing in Tunnels.....	7
3.3 IPv6 Decapsulation.....	7
3.4 IPv6 Tunnel Protocol Engine.....	8
4. Nested Encapsulation.....	11
4.1 Limiting Nested Encapsulation.....	12
4.1.1 Tunnel Encapsulation Limit Option.....	13
4.1.2 Loopback Encapsulation.....	15
4.1.3 Routing Loop Nested Encapsulation.....	15
5. Tunnel IPv6 Header.....	16
5.1 Tunnel IPv6 Extension Headers.....	17
6. IPv6 Tunnel State Variables.....	19
6.1 IPv6 Tunnel Entry-Point Node.....	19
6.2 IPv6 Tunnel Exit-Point Node.....	19

6.3 IPv6 Tunnel Hop Limit.....	19
6.4 IPv6 Tunnel Packet Traffic Class.....	20
6.5 IPv6 Tunnel Flow Label.....	20
6.6 IPv6 Tunnel Encapsulation Limit.....	20
6.7 IPv6 Tunnel MTU.....	20
7. IPv6 Tunnel Packet Size Issues.....	21
7.1 IPv6 Tunnel Packet Fragmentation.....	21
7.2 IPv4 Tunnel Packet Fragmentation.....	22
8. IPv6 Tunnel Error Reporting and Processing.....	22
8.1 Tunnel ICMP Messages.....	27
8.2 ICMP Messages for IPv6 Original Packets.....	28
8.3 ICMP Messages for IPv4 Original Packets.....	29
8.4 ICMP Messages for Nested Tunnel Packets.....	30
9. Security Considerations.....	30
10. Acknowledgments.....	31
11. References.....	31
Authors' Addresses.....	32
Appendix A. Risk Factors in Recursive Encapsulation.....	33
Full Copyright Statement.....	36

1. Introduction

This document specifies a method and generic mechanisms by which a packet is encapsulated and carried as payload within an IPv6 packet. The resulting packet is called an IPv6 tunnel packet. The forwarding path between the source and destination of the tunnel packet is called an IPv6 tunnel. The technique is called IPv6 tunneling.

A typical scenario for IPv6 tunneling is the case in which an intermediate node exerts explicit routing control by specifying particular forwarding paths for selected packets. This control is achieved by prepending IPv6 headers to each of the selected original packets. These prepended headers identify the forwarding paths.

In addition to the description of generic IPv6 tunneling mechanisms, which is the focus of this document, specific mechanisms for tunneling IPv6 and IPv4 packets are also described herein.

The keywords MUST, MUST NOT, MAY, OPTIONAL, REQUIRED, RECOMMENDED, SHALL, SHALL NOT, SHOULD, SHOULD NOT are to be interpreted as defined in RFC 2119.

2. Terminology

original packet

a packet that undergoes encapsulation.

original header

the header of an original packet.

tunnel

a forwarding path between two nodes on which the payloads of packets are original packets.

tunnel end-node

a node where a tunnel begins or ends.

tunnel header

the header prepended to the original packet during encapsulation. It specifies the tunnel end-points as source and destination.

tunnel packet

a packet that encapsulates an original packet.

tunnel entry-point

the tunnel end-node where an original packet is encapsulated.

tunnel exit-point

the tunnel end-node where a tunnel packet is decapsulated.

IPv6 tunnel

a tunnel configured as a virtual link between two IPv6 nodes, on which the encapsulating protocol is IPv6.

tunnel MTU

the maximum size of a tunnel packet payload without requiring fragmentation, that is, the Path MTU between the tunnel entry-point and the tunnel exit-point nodes minus the size of the tunnel header.

tunnel hop limit

the maximum number of hops that a tunnel packet can travel from the tunnel entry-point to the tunnel exit-point.

inner tunnel

a tunnel that is a hop (virtual link) of another tunnel.

outer tunnel

a tunnel containing one or more inner tunnels.

nested tunnel packet

a tunnel packet that has as payload a tunnel packet.

nested tunnel header

the tunnel header of a nested tunnel packet.

nested encapsulation

encapsulation of an encapsulated packet.

recursive encapsulation

encapsulation of a packet that reenters a tunnel before exiting it.

tunnel encapsulation limit

the maximum number of nested encapsulations of a packet.

3. IPv6 Tunneling

IPv6 tunneling is a technique for establishing a "virtual link" between two IPv6 nodes for transmitting data packets as payloads of IPv6 packets (see Fig.1). From the point of view of the two nodes, this "virtual link", called an IPv6 tunnel, appears as a point to point link on which IPv6 acts like a link-layer protocol. The two IPv6 nodes play specific roles. One node encapsulates original packets received from other nodes or from itself and forwards the resulting tunnel packets through the tunnel. The other node decapsulates the received tunnel packets and forwards the resulting original packets towards their destinations, possibly itself. The encapsulator node is called the tunnel entry-point node, and it is the source of the tunnel packets. The decapsulator node is called the tunnel exit-point, and it is the destination of the tunnel packets.

Note:

This document refers in particular to tunnels between two nodes identified by unicast addresses - such tunnels look like "virtual point to point links". The mechanisms described herein apply also to tunnels in which the exit-point nodes are identified by other types of addresses, such as anycast or multicast. These tunnels may look like "virtual point to multipoint links". At the time of writing this document, IPv6 anycast addresses are a subject of ongoing specification and experimental work.

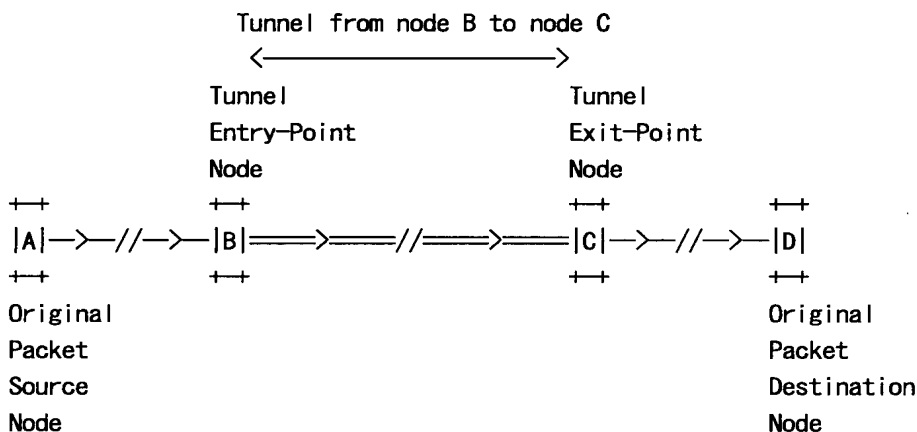


Fig. 1 Tunnel

An IPv6 tunnel is a unidirectional mechanism – tunnel packet flow takes place in one direction between the IPv6 tunnel entry-point and exit-point nodes (see Fig.1).

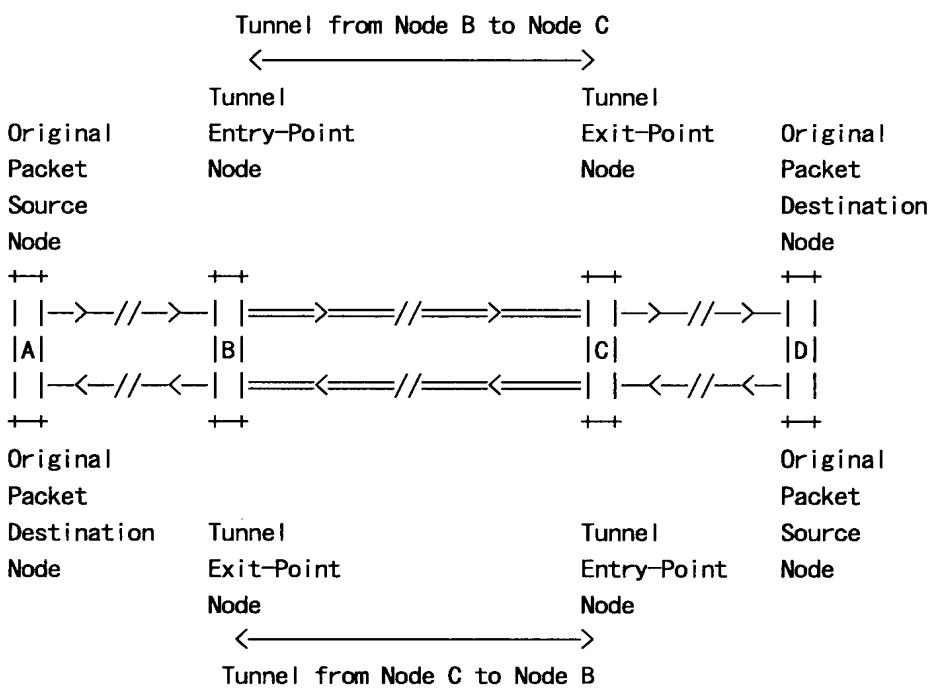


Fig.2 Bi-directional Tunneling Mechanism

Bi-directional tunneling is achieved by merging two unidirectional mechanisms, that is, configuring two tunnels, each in opposite direction to the other - the entry-point node of one tunnel is the exit-point node of the other tunnel (see Fig.2).

3.1 IPv6 Encapsulation

IPv6 encapsulation consists of prepending to the original packet an IPv6 header and, optionally, a set of IPv6 extension headers (see Fig.3), which are collectively called tunnel IPv6 headers. The encapsulation takes place in an IPv6 tunnel entry-point node, as the result of an original packet being forwarded onto the virtual link represented by the tunnel. The original packet is processed during forwarding according to the forwarding rules of the protocol of that packet. For instance if the original packet is an:

- (a) IPv6 packet, the IPv6 original header hop limit is decremented by one.
- (b) IPv4 packet, the IPv4 original header time to live field (TTL) is decremented by one.

At encapsulation, the source field of the tunnel IPv6 header is filled with an IPv6 address of the tunnel entry-point node, and the destination field with an IPv6 address of the tunnel exit-point. Subsequently, the tunnel packet resulting from encapsulation is sent towards the tunnel exit-point node.

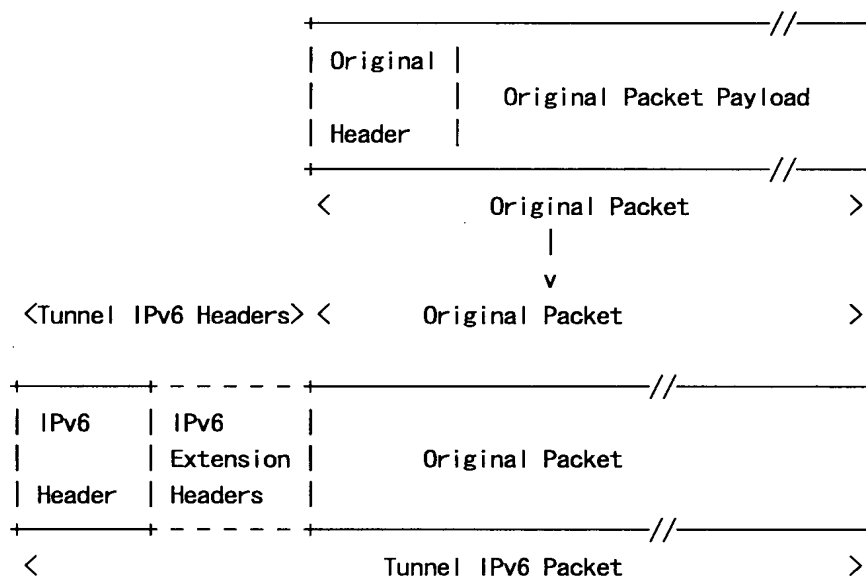


Fig. 3 Encapsulating a Packet

Tunnel extension headers should appear in the order recommended by the specifications that define the extension headers, such as [IPv6-Spec].

A source of original packets and a tunnel entry-point that encapsulates those packets can be the same node.

3.2 Packet Processing in Tunnels

The intermediate nodes in the tunnel process the IPv6 tunnel packets according to the IPv6 protocol. For example, a tunnel Hop by Hop Options extension header is processed by each receiving node in the tunnel; a tunnel Routing extension header identifies the intermediate processing nodes, and controls at a finer granularity the forwarding path of the tunnel packet through the tunnel; a tunnel Destination Options extension header is processed at the tunnel exit-point node.

3.3 IPv6 Decapsulation

Decapsulation is graphically shown in Fig.4:

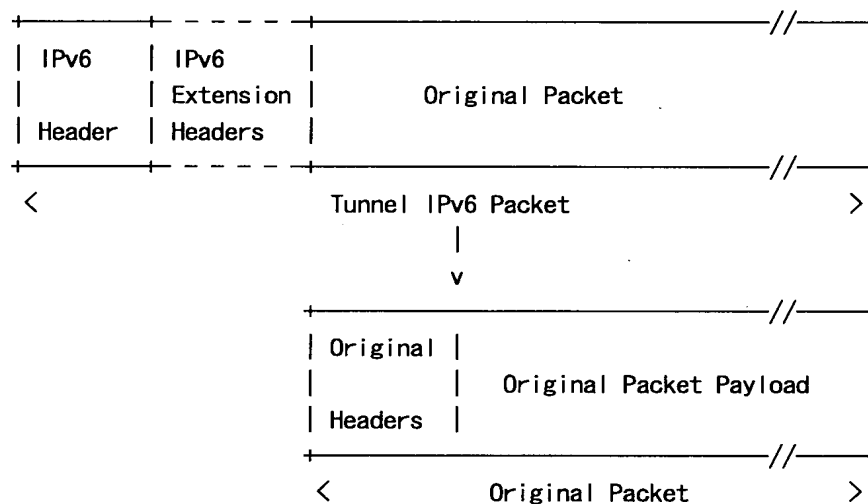


Fig.4 Decapsulating a Packet

Upon receiving an IPv6 packet destined to an IPv6 address of a tunnel exit-point node, its IPv6 protocol layer processes the tunnel headers. The strict left-to-right processing rules for extension headers is applied. When processing is complete, control is handed to the next protocol engine, which is identified by the Next Header field value in the last header processed. If this is set to a tunnel protocol value, the tunnel protocol engine discards the tunnel headers and passes the resulting original packet to the Internet or lower layer protocol identified by that value for further processing.

For example, in the case the Next Header field has the IPv6 Tunnel Protocol value, the resulting original packet is passed to the IPv6 protocol layer.

The tunnel exit-point node, which decapsulates the tunnel packets, and the destination node, which receives the resulting original packets can be the same node.

3.4 IPv6 Tunnel Protocol Engine

Packet flow (paths #1-7) through the IPv6 Tunnel Protocol Engine on a node is graphically shown in Fig.5:

Note:

In Fig.5, the Upper-Layer Protocols box represents transport protocols such as TCP, UDP, control protocols such as ICMP, routing protocols such as OSPF, and internet or lower-layer protocol being "tunneled" over IPv6, such as IPv4, IPX, etc. The Link-Layer Protocols box represents Ethernet, Token Ring, FDDI, PPP, X.25, Frame Relay, ATM, etc..., as well as internet layer "tunnels" such as IPv4 tunnels.

The IPv6 tunnel protocol engine acts as both an "upper-layer" and a "link-layer", each with a specific input and output as follows:

(u.i) "tunnel upper-layer input" - consists of tunnel IPv6 packets that are going to be decapsulated. The tunnel packets are incoming through the IPv6 layer from:

(u.i.1) a link-layer - (path #1, Fig.5)

These are tunnel packets destined to this node and will undergo decapsulation.

(u.i.2) a tunnel link-layer - (path #7, Fig.5)

These are tunnel packets that underwent one or more decapsulations on this node, that is, the packets had one or more nested tunnel headers and one nested tunnel header was just discarded. This node is the exit-point of both an outer tunnel and one or more of its inner tunnels.

For both above cases the resulting original packets are passed back to the IPv6 layer as "tunnel link-layer" output for further processing (see b.2).

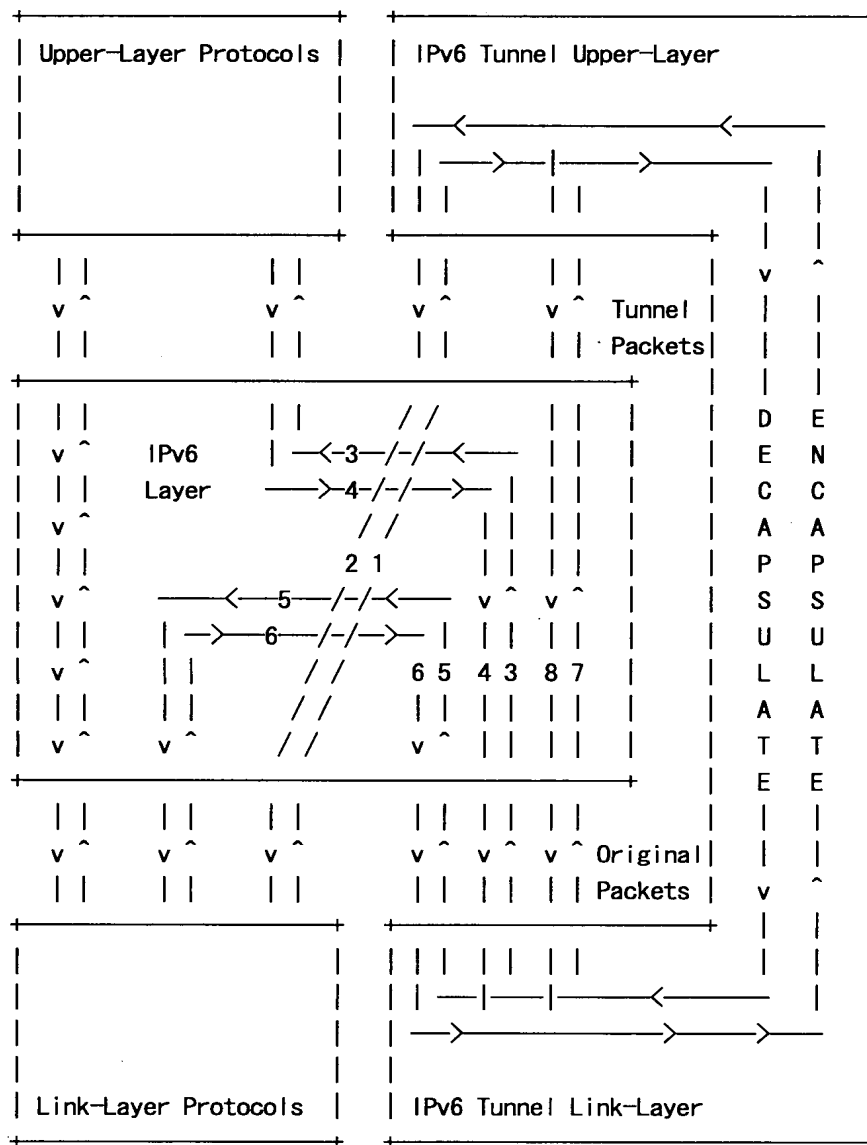


Fig.5 Packet Flow in the IPv6 Tunneling Protocol Engine on a Node

(u.o) "tunnel upper-layer output" - consists of tunnel IPv6 packets that are passed through the IPv6 layer down to:

(u.o.1) a link-layer - (path #2, Fig. 5)

These packets underwent encapsulation and are sent towards the tunnel exit-point

(u.o.2) a tunnel link-layer - (path #8, Fig.5)

These tunnel packets undergo nested encapsulation. This node is the entry-point node of both an outer tunnel and one or more of its inner tunnel.

Implementation Note:

The tunnel upper-layer input and output can be implemented similar to the input and output of the other upper-layer protocols.

The tunnel link-layer input and output are as follows:

- (l.i) "tunnel link-layer input" - consists of original IPv6 packets that are going to be encapsulated.

The original packets are incoming through the IPv6 layer from:

- (l.i.1) an upper-layer - (path #4, Fig.5)

These are original packets originating on this node that undergo encapsulation. The original packet source and tunnel entry-point are the same node.

- (l.i.2) a link-layer - (path #6, Fig.5)

These are original packets incoming from a different node that undergo encapsulation on this tunnel entry-point node.

- (l.i.3) a tunnel upper-layer - (path #8, Fig.5)

These packets are tunnel packets that undergo nested encapsulation. This node is the entry-point node of both an outer tunnel and one or more of its inner tunnels.

The resulting tunnel packets are passed as tunnel upper-layer output packets through the IPv6 layer (see u.o) down to:

- (l.o) "tunnel link-layer output" - consists of original IPv6 packets resulting from decapsulation. These packets are passed through the IPv6 layer to:

- (l.o.1) an upper-layer - (path #3, Fig.5)

These original packets are destined to this node.

- (l.o.2) a link-layer - (path #5, Fig.5)

These original packets are destined to another node; they are transmitted on a link towards their destination.

(l.o.3) a tunnel upper-layer - (path #7, Fig.5)

These packets undergo another decapsulation; they were nested tunnel packets. This node is both the exit-point node of an outer tunnel and one or more inner tunnels.

Implementation Note:

The tunnel link-layer input and output can be implemented similar to the input and output of other link-layer protocols, for instance, associating an interface or pseudo-interface with the IPv6 tunnel.

The selection of the "IPv6 tunnel link" over other links results from the packet forwarding decision taken based on the content of the node's routing table.

4. Nested Encapsulation

Nested IPv6 encapsulation is the encapsulation of a tunnel packet. It takes place when a hop of an IPv6 tunnel is a tunnel. The tunnel containing a tunnel is called an outer tunnel. The tunnel contained in the outer tunnel is called an inner tunnel - see Fig.6. Inner tunnels and their outer tunnels are nested tunnels.

The entry-point node of an "inner IPv6 tunnel" receives tunnel IPv6 packets encapsulated by the "outer IPv6 tunnel" entry-point node. The "inner tunnel entry-point node" treats the receiving tunnel packets as original packets and performs encapsulation. The resulting packets are "tunnel packets" for the "inner IPv6 tunnel", and "nested tunnel packets" for the "outer IPv6 tunnel".

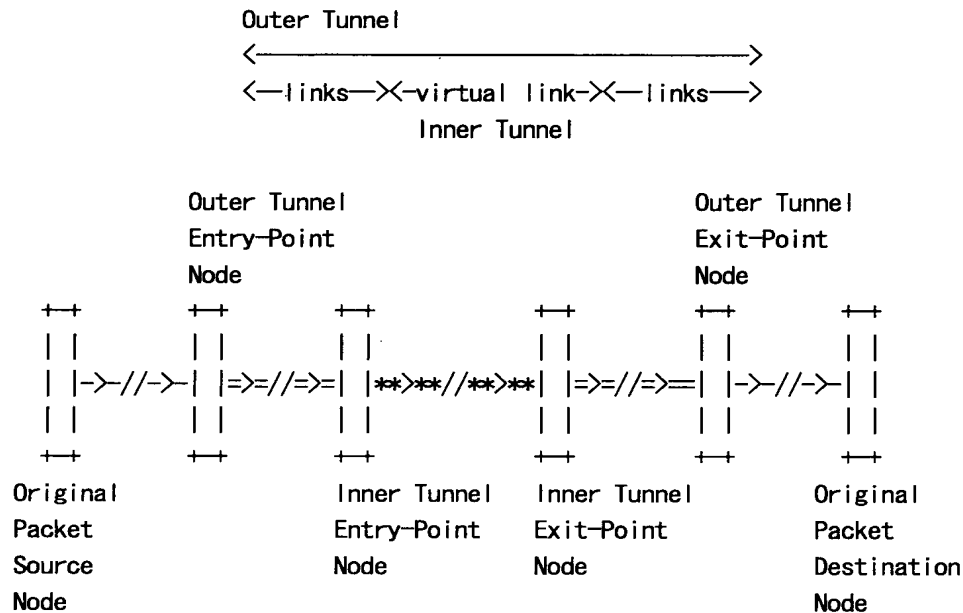


Fig. 6. Nested Encapsulation

4.1 Limiting Nested Encapsulation

A tunnel IPv6 packet is limited to the maximum IPv6 packet size [IPv6-Spec]. Each encapsulation adds to the size of an encapsulated packet the size of the tunnel IPv6 headers. Consequently, the number of tunnel headers, and therefore, the number of nested encapsulations is limited by the maximum packet size. However this limit is so large (more than 1600 encapsulations for an original packet of minimum size) that it is not an effective limit in most cases.

The increase in the size of a tunnel IPv6 packet due to nested encapsulations may require fragmentation [IPv6-Spec] at a tunnel entry point - see section 7. Furthermore, each fragmentation, due to nested encapsulation, of an already fragmented tunnel packet results in a doubling of the number of fragments. Moreover, it is probable that once this fragmentation begins, each new nested encapsulation results in yet additional fragmentation. Therefore limiting nested encapsulation is recommended.

The proposed mechanism for limiting excessive nested encapsulation is a "Tunnel Encapsulation Limit" option, which is carried in an IPv6 Destination Options extension header accompanying an encapsulating IPv6 header.

4.1.1 Tunnel Encapsulation Limit Option

A tunnel entry-point node may be configured to include a Tunnel Encapsulation Limit option as part of the information prepended to all packets entering a tunnel at that node. The Tunnel Encapsulation Limit option is carried in a Destination Options extension header [IPv6-Spec] placed between the encapsulating IPv6 header and the IPv6 header of the original packet. (Other IPv6 extension headers may also be present preceding or following the Destination Options extension header, depending on configuration information at the tunnel entry-point node.)

The Tunnel Encapsulation Limit option specifies how many additional levels of encapsulation are permitted to be prepended to the packet — or, in other words, how many further levels of nesting the packet is permitted to undergo — not counting the encapsulation in which the option itself is contained. For example, a Tunnel Encapsulation Limit option containing a limit value of zero means that a packet carrying that option may not enter another tunnel before exiting the current tunnel.

The Tunnel Encapsulation Limit option has the following format:

Option Type	Opt Data Len	Opt Data Len
0 1 2 3 4 5 6 7		
0 0 0 0 0 1 0 0	1	Tun Encap Lim

Option Type decimal value 4

- the highest-order two bits - set to 00 - indicate "skip over this option if the option is not recognized".

- the third-highest-order bit - set to 0 - indicates that the option data in this option does not change en route to the packet's destination [IPv6-Spec].

Opt Data Len value 1 - the data portion of the Option is one octet long.

Opt Data Value the Tunnel Encapsulation Limit value - 8-bit unsigned integer specifying how many further levels of encapsulation are permitted for the

Tunnel Encapsulation Limit options are of interest only to tunnel entry points. A tunnel entry-point node is required to execute the following procedure for every packet entering a tunnel at that node:

- (a) Examine the packet to see if a Tunnel Encapsulation Limit option is present following its IPv6 header. The headers following the IPv6 header must be examined in strict "left-to-right" order, with the examination stopping as soon as any one of the following headers is encountered:
 - (i) a Destination Options extension header containing a Tunnel Encapsulation Limit, (ii) another IPv6 header, (iii) a non-extension header, such as TCP, UDP, or ICMP, or (iv) a header that cannot be parsed because it is encrypted or its type is unknown. (Note that this requirement is an exception to the general IPv6 rule that a Destination Options extension header need only be examined by a packet's destination node. An alternative and "cleaner" approach would have been to use a Hop-by-Hop extension header for this purpose, but that would have imposed an undesirable extra processing burden, and possible consequent extra delay, at every IPv6 node along the path of a tunnel.)
- (b) If a Tunnel Encapsulation Limit option is found in the packet entering the tunnel and its limit value is zero, the packet is discarded and an ICMP Parameter Problem message [ICMP-Spec] is sent to the source of the packet, which is the previous tunnel entry-point node. The Code field of the Parameter Problem message is set to zero ("erroneous header field encountered") and the Pointer field is set to point to the third octet of the Tunnel Encapsulation Limit option (i.e., the octet containing the limit value of zero).
- (c) If a Tunnel Encapsulation Limit option is found in the packet entering the tunnel and its limit value is non-zero, an additional Tunnel Encapsulation Limit option must be included as part of the encapsulating headers being added at this entry point. The limit value in the encapsulating option is set to one less than the limit value found in the packet being encapsulated.
- (d) If a Tunnel Encapsulation Limit option is not found in the packet entering the tunnel and if an encapsulation limit has been configured for this tunnel, a Tunnel Encapsulation Limit option must be included as part of the encapsulating headers being added at this entry point. The limit value in the option is set to the configured limit.

- (e) If a Tunnel Encapsulation Limit option is not found in the packet entering the tunnel and if no encapsulation limit has been configured for this tunnel, then no Tunnel Encapsulation Limit option is included as part of the encapsulating headers being added at this entry point.

A Tunnel Encapsulation Limit option added at a tunnel entry-point node is removed as part of the decapsulation process at that tunnel's exit-point node.

Two cases of encapsulation that should be avoided are described below:

4.1.2 Loopback Encapsulation

A particular case of encapsulation which must be avoided is the loopback encapsulation. Loopback encapsulation takes place when a tunnel IPv6 entry-point node encapsulates tunnel IPv6 packets originated from itself, and destined to itself. This can generate an infinite processing loop in the entry-point node.

To avoid such a case, it is recommended that an implementation have a mechanism that checks and rejects the configuration of a tunnel in which both the entry-point and exit-point node addresses belong to the same node. It is also recommended that the encapsulating engine check for and reject the encapsulation of a packet that has the pair of tunnel entry-point and exit-point addresses identical with the pair of original packet source and final destination addresses.

4.1.3 Routing-Loop Nested Encapsulation

In the case of a forwarding path with multiple-level nested tunnels, a routing-loop from an inner tunnel to an outer tunnel is particularly dangerous when packets from the inner tunnels reenter an outer tunnel from which they have not yet exited. In such a case, the nested encapsulation becomes a recursive encapsulation with the negative effects described in 4.1. Because each nested encapsulation adds a tunnel header with a new hop limit value, the IPv6 hop limit mechanism cannot control the number of times the packet reaches the outer tunnel entry-point node, and thus cannot control the number of recursive encapsulations.

When the path of a packet from source to final destination includes tunnels, the maximum number of hops that the packet can traverse should be controlled by two mechanisms used together to avoid the negative effects of recursive encapsulation in routing loops:

- (a) the original packet hop limit.

It is decremented at each forwarding operation performed on an original packet. This includes each encapsulation of the original packet. It does not include nested encapsulations of the original packet

- (b) the tunnel IPv6 packet encapsulation limit.

It is decremented at each nested encapsulation of the packet.

For a discussion of the excessive encapsulation risk factors in nested encapsulation see Appendix A.

5. Tunnel IPv6 Header

The tunnel entry-point node fills out a tunnel IPv6 main header [IPv6-Spec] as follows:

Version:

value 6

Traffic Class:

Depending on the entry-point node tunnel configuration, the traffic class can be set to that of either the original packet or a pre-configured value - see section 6.4.

Flow Label:

Depending on the entry-point node tunnel configuration, the flow label can be set to a pre-configured value. The typical value is zero - see section 6.5.

Payload Length:

The original packet length, plus the length of the encapsulating (prepended) IPv6 extension headers, if any.

Next Header:

The next header value according to [IPv6-Spec] from the Assigned Numbers RFC [RFC-1700 or its successors].

For example, if the original packet is an IPv6 packet, this is set to:

- decimal value 41 (Assigned Next Header number for IPv6) - if there are no tunnel extension headers.
- value 0 (Assigned Next Header number for IPv6 Hop by Hop Options extension header) - if a hop by hop options extension header immediately follows the tunnel IPv6 header.
- decimal value 60 (Assigned Next Header number for IPv6 Destination Options extension header) - if a destination options extension header immediately follows the tunnel IPv6 header.

Hop Limit:

The tunnel IPv6 header hop limit is set to a pre-configured value - see section 6.3.

The default value for hosts is the Neighbor Discovery advertised hop limit [ND-Spec]. The default value for routers is the default IPv6 Hop Limit value from the Assigned Numbers RFC (64 at the time of writing this document).

Source Address:

An IPv6 address of the outgoing interface of the tunnel entry-point node. This address is configured as the tunnel entry-point node address - see section 6.1.

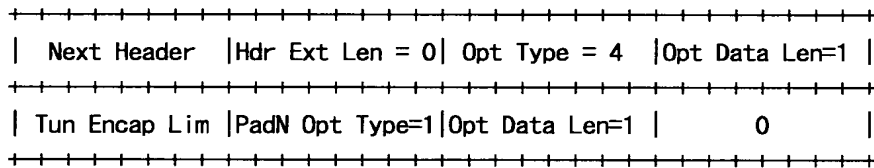
Destination Address:

An IPv6 address of the tunnel exit-point node. This address is configured as the tunnel exit-point node address - see section 6.2.

5.1 Tunnel IPv6 Extension Headers

Depending on IPv6 node configuration parameters, a tunnel entry-point node may append to the tunnel IPv6 main header one or more IPv6 extension headers, such as a Hop-by-Hop Options header, a Routing header, or others.

To limit the number of nested encapsulations of a packet, if it was configured to do so - see section 6.6 - a tunnel entry-point includes a Destination Options extension header containing a Tunnel Encapsulation Limit option. If that option is the only option present in the Destination Options header, the header has the following format:



Next Header:

Identifies the type of the original packet header. For example, if the original packet is an IPv6 packet, the next header protocol value is set to decimal value 41 (Assigned payload type number for IPv6).

Hdr Ext Len:

Length of the Destination Options extension header in 8-octet units, not including the first 8 octets. Set to value 0, if no other options are present in this destination options header.

Option Type:

value 4 - see section 4.1.1.

Opt Data Len:

value 1 - see section 4.1.1.

Tun Encap Lim:

8 bit unsigned integer - see section 4.1.1.

Option Type:

value 1 - PadN option, to align the header following this header.

Opt Data Len:

value 1 - one octet of option data.

Option Data:

value 0 – one zero-valued octet.

6. IPv6 Tunnel State Variables

The IPv6 tunnel state variables, some of which are or may be configured on the tunnel entry-point node, are:

6.1 IPv6 Tunnel Entry-Point Node Address

The tunnel entry-point node address is one of the valid IPv6 unicast addresses of the entry-point node – the validation of the address at tunnel configuration time is recommended.

The tunnel entry-point node address is copied to the source address field in the tunnel IPv6 header during packet encapsulation.

6.2 IPv6 Tunnel Exit-Point Node Address

The tunnel exit-point node address is used as IPv6 destination address for the tunnel IPv6 header. A tunnel acts like a virtual point to point link between the entry-point node and exit-point node.

The tunnel exit-point node address is copied to the destination address field in the tunnel IPv6 header during packet encapsulation.

The configuration of the tunnel entry-point and exit-point addresses is not subject to IPv6 Autoconfiguration or IPv6 Neighbor Discovery.

6.3 IPv6 Tunnel Hop Limit

An IPv6 tunnel is modeled as a “single-hop virtual link” tunnel, in which the passing of the original packet through the tunnel is like the passing of the original packet over a one hop link, regardless of the number of hops in the IPv6 tunnel.

The “single-hop” mechanism should be implemented by having the tunnel entry point node set a tunnel IPv6 header hop limit independently of the hop limit of the original header.

The “single-hop” mechanism hides from the original IPv6 packets the number of IPv6 hops of the tunnel.

It is recommended that the tunnel hop limit be configured with a value that ensures:

- (a) that tunnel IPv6 packets can reach the tunnel exit-point node
- (b) a quick expiration of the tunnel packet if a routing loop occurs within the IPv6 tunnel.

The tunnel hop limit default value for hosts is the IPv6 Neighbor Discovery advertised hop limit [ND-Spec]. The tunnel hop limit default value for routers is the default IPv6 Hop Limit value from the Assigned Numbers RFC (64 at the time of writing this document).

The tunnel hop limit is copied into the hop limit field of the tunnel IPv6 header of each packet encapsulated by the tunnel entry-point node.

6.4 IPv6 Tunnel Packet Traffic Class

The IPv6 Tunnel Packet Traffic Class indicates the value that a tunnel entry-point node sets in the Traffic Class field of a tunnel header. The default value is zero. The configured Packet Traffic Class can also indicate whether the value of the Traffic Class field in the tunnel header is copied from the original header, or it is set to the pre-configured value.

6.5 IPv6 Tunnel Flow Label

The IPv6 Tunnel Flow Label indicates the value that a tunnel entry-point node sets in the flow label of a tunnel header. The default value is zero.

6.6 IPv6 Tunnel Encapsulation Limit

The Tunnel Encapsulation Limit value can indicate whether the entry-point node is configured to limit the number of encapsulations of tunnel packets originating on that node. The IPv6 Tunnel Encapsulation Limit is the maximum number of additional encapsulations permitted for packets undergoing encapsulation at that entry-point node. Recommended default value is 4. An entry-point node configured to limit the number of nested encapsulations prepends a Destination Options extension header containing a Tunnel Encapsulation Limit option to an original packet undergoing encapsulation - see sections 4.1 and 4.1.1.

6.7 IPv6 Tunnel MTU

The tunnel MTU is set dynamically to the Path MTU between the tunnel entry-point and the tunnel exit-point nodes, minus the size of the tunnel headers: the maximum size of a tunnel packet payload that can

be sent through the tunnel without fragmentation [IPv6-Spec]. The tunnel entry-point node performs Path MTU discovery on the path between the tunnel entry-point and exit-point nodes [PMTU-Spec], [ICMP-Spec]. The tunnel MTU of a nested tunnel is the tunnel MTU of the outer tunnel minus the size of the nested tunnel headers.

7. IPv6 Tunnel Packet Size Issues

Prepending a tunnel header increases the size of a packet, therefore a tunnel packet resulting from the encapsulation of an IPv6 original packet may require fragmentation.

A tunnel IPv6 packet resulting from the encapsulation of an original packet is considered an IPv6 packet originating from the tunnel entry-point node. Therefore, like any source of an IPv6 packet, a tunnel entry-point node must support fragmentation of tunnel IPv6 packets.

A tunnel intermediate node that forwards a tunnel packet to another node in the tunnel follows the general IPv6 rule that it must not fragment a packet undergoing forwarding.

A tunnel exit-point node receiving tunnel packets at the end of the tunnel for decapsulation applies the strict left-to-right processing rules for extension headers. In the case of a fragmented tunnel packet, the fragments are reassembled into a complete tunnel packet before determining that an embedded packet is present.

Note:

A particular problem arises when the destination of a fragmented tunnel packet is an exit-point node identified by an anycast address. The problem, which is similar to that of original fragmented IPv6 packets destined to nodes identified by an anycast address, is that all the fragments of a packet must arrive at the same destination node for that node to be able to perform a successful reassembly, a requirement that is not necessarily satisfied by packets sent to an anycast address.

7.1 IPv6 Tunnel Packet Fragmentation

When an IPv6 original packet enters a tunnel, if the original packet size exceeds the tunnel MTU (i.e., the Path MTU between the tunnel entry-point and the tunnel exit-point, minus the size of the tunnel header(s)), it is handled as follows:

- (a) if the original IPv6 packet size is larger than the IPv6 minimum link MTU [IPv6-Spec], the entry-point node discards the packet and sends an ICMPv6 "Packet Too Big" message to the source address of the original packet with the recommended MTU size field set to the tunnel MTU or the IPv6 minimum link MTU, whichever is larger, i.e. max (tunnel MTU, IPv6 minimum link MTU). Also see sections 6.7 and 8.2.
- (b) if the original IPv6 packet is equal or smaller than the IPv6 minimum link MTU, the tunnel entry-point node encapsulates the original packet, and subsequently fragments the resulting IPv6 tunnel packet into IPv6 fragments that do not exceed the Path MTU to the tunnel exit-point.

7.2 IPv4 Tunnel Packet Fragmentation

When an IPv4 original packet enters a tunnel, if the original packet size exceeds the tunnel MTU (i.e., the Path MTU between the tunnel entry-point and the tunnel exit-point, minus the size of the tunnel header(s)), it is handled as follows:

- (a) if in the original IPv4 packet header the Don't Fragment - DF - bit flag is SET, the entry-point node discards the packet and returns an ICMP message. The ICMP message has the type = "unreachable", the code = "packet too big", and the recommended MTU size field set to the size of the tunnel MTU - see sections 6.7 and 8.3.
- (b) if in the original packet header the Don't Fragment - DF - bit flag is CLEAR, the tunnel entry-point node encapsulates the original packet, and subsequently fragments the resulting IPv6 tunnel packet into IPv6 fragments that do not exceed the Path MTU to the tunnel exit-point.

8. IPv6 Tunnel Error Processing and Reporting

IPv6 Tunneling follows the general rule that an error detected during the processing of an IPv6 packet is reported through an ICMP message to the source of the packet.

On a forwarding path that includes IPv6 tunnels, an error detected by a node that is not in any tunnel is directly reported to the source of the original IPv6 packet.

An error detected by a node inside a tunnel is reported to the source of the tunnel packet, that is, the tunnel entry-point node. The ICMP message sent to the tunnel entry-point node has as ICMP payload the tunnel IPv6 packet that has the original packet as its payload.

The cause of a packet error encountered inside a tunnel can be a problem with:

- (a) the tunnel header, or
- (b) the tunnel packet.

Both tunnel header and tunnel packet problems are reported to the tunnel entry-point node.

If a tunnel packet problem is a consequence of a problem with the original packet, which is the payload of the tunnel packet, then the problem is also reported to the source of the original packet.

To report a problem detected inside the tunnel to the source of an original packet, the tunnel entry point node must relay the ICMP message received from inside the tunnel to the source of that original IPv6 packet.

An example of the processing that can take place in the error reporting mechanism of a node is illustrated in Fig.7, and Fig.8:

Fig.7 path #0 and Fig.8 (a) - The IPv6 tunnel entry-point receives an ICMP packet from inside the tunnel, marked Tunnel ICMPv6 Message in Fig.7. The tunnel entry-point node IPv6 layer passes the received ICMP message to the ICMPv6 Input. The ICMPv6 Input, based on the ICMP type and code [ICMP-Spec] generates an internal "error code".

Fig.7 path #1 - The internal error code, is passed with the "ICMPv6 message payload" to the upper-layer protocol - in this case the IPv6 tunnel upper-layer error input.

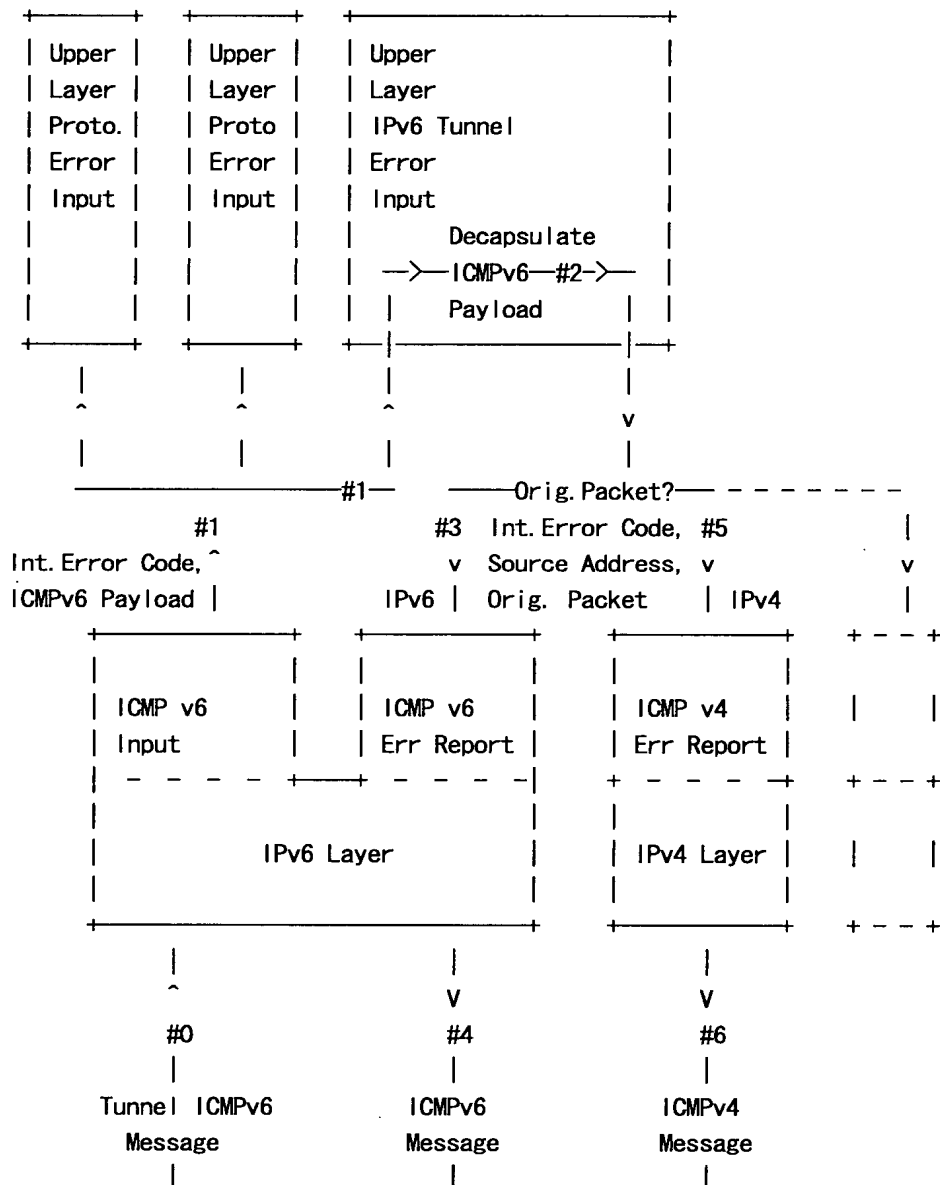


Fig. 7 Error Reporting Flow in a Node (IPv6 Tunneling Protocol Engine)

Fig. 7 path #2 and Fig. 8 (b) – The IPv6 tunnel error input decapsulates the tunnel IPv6 packet, which is the ICMPv6 message payload, obtaining the original packet, and thus the original headers and dispatches the “internal error code”, the source address from the original packet header, and the original packet, down to the error report block of the protocol identified by the Next Header field in the tunnel header immediately preceding the original packet in the ICMP message payload.

From here the processing depends on the protocol of the original packet:

(a) - for an IPv6 original packet

Fig. 7 path #3 and Fig. 8 (c. 1)- for an IPv6 original packet, the ICMPv6 error report builds an ICMP message of a type and code according to the "internal error code", containing the "original packet" as ICMP payload.

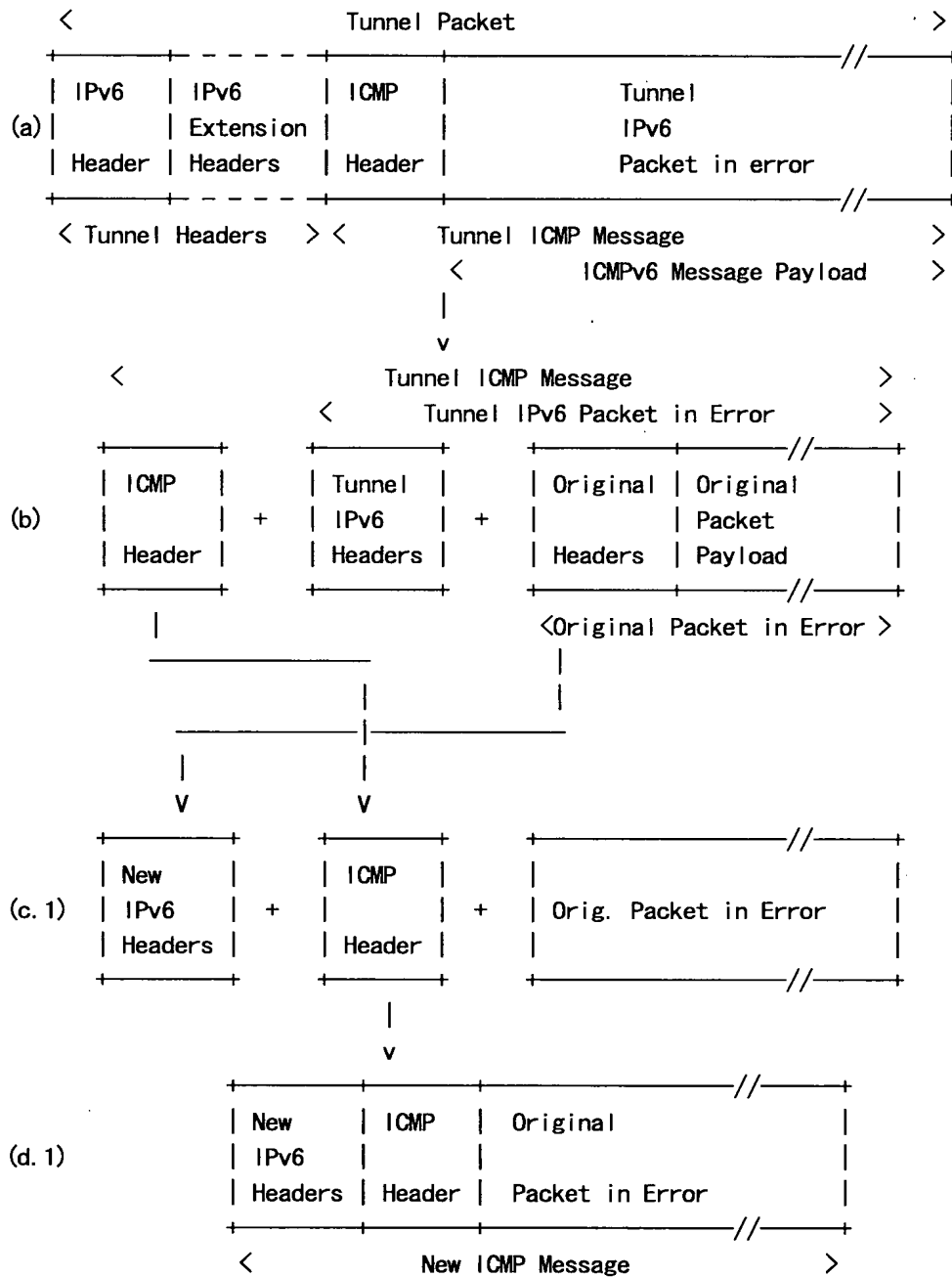
Fig. 7 path #4 and Fig. 8 (d. 1)- The ICMP message has the tunnel entry-point node address as source address, and the original packet source node address as destination address. The tunnel entry-point node sends the ICMP message to the source node of the original packet.

(b) - for an IPv4 original packet

Fig. 7 path #5 and Fig. 8 (c. 2) - for an IPv4 original packet, the ICMPv4 error report builds an ICMP message of a type and code derived from the the "internal error code", containing the "original packet" as ICMP payload.

Fig. 7 path #6 and Fig. 8 (d. 2) - The ICMP message has the tunnel entry-point node IPv4 address as source address, and the original packet IPv4 source node address as destination address. The tunnel entry-point node sends the ICMP message to the source node of the original packet.

A graphical description of the header processing taking place is the following:



or for an IPv4 original packet

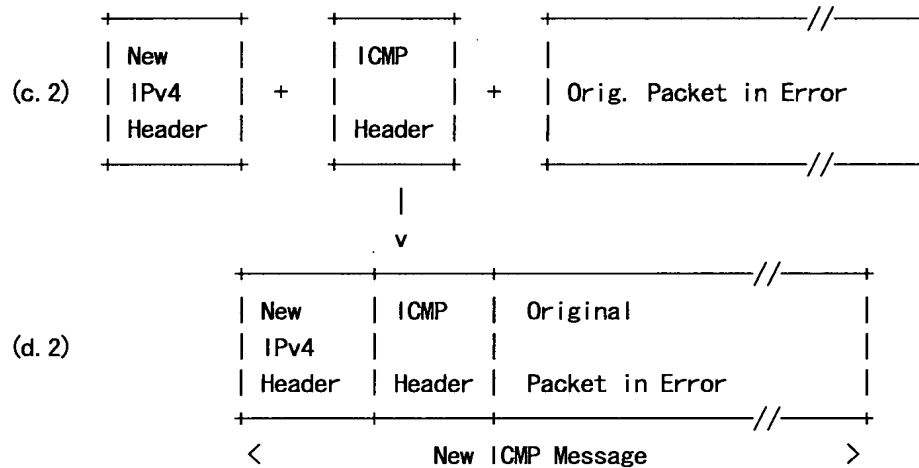


Fig. 8 ICMP Error Reporting and Processing

8.1 Tunnel ICMP Messages

The tunnel ICMP messages that are reported to the source of the original packet are:

hop limit exceeded

The tunnel has a misconfigured hop limit, or contains a routing loop, and packets do not reach the tunnel exit-point node. This problem is reported to the tunnel entry-point node, where the tunnel hop limit can be reconfigured to a higher value. The problem is further reported to the source of the original packet as described in section 8.2, or 8.3.

unreachable node

One of the nodes in the tunnel is not or is no longer reachable. This problem is reported to the tunnel entry-point node, which should be reconfigured with a valid and active path between the entry and exit-point of the tunnel.

The problem is further reported to the source of the original packet as described in section 8.2, or 8.3.

parameter problem

A Parameter Problem ICMP message pointing to a valid Tunnel Encapsulation Limit Destination header with a Tun Encap Lim field value set to one is an indication that the tunnel

packet exceeded the maximum number of encapsulations allowed. The problem is further reported to the source of the original packet as described in section 8.2, or 8.3.

The above three problems detected inside the tunnel, which are a tunnel configuration and a tunnel topology problem, are reported to the source of the original IPv6 packet, as a tunnel generic "unreachable" problem caused by a "link problem" - see section 8.2 and 8.3.

packet too big

The tunnel packet exceeds the tunnel Path MTU.

The information carried by this type of ICMP message is used as follows:

- by a receiving tunnel entry-point node to set or adjust the tunnel MTU
- by a sending tunnel entry-point node to indicate to the source of an original packet the MTU size that should be used in sending IPv6 packets towards the tunnel entry-point node.

8.2 ICMP Messages for IPv6 Original Packets

The tunnel entry-point node builds the ICMP and IPv6 headers of the ICMP message that is sent to the source of the original packet as follows:

IPv6 Fields:

Source Address

A valid unicast IPv6 address of the outgoing interface.

Destination Address

Copied from the Source Address field of the Original IPv6 header.

ICMP Fields:

For any of the following tunnel ICMP error messages:

"hop limit exceeded"

"unreachable node"

"parameter problem" - pointing to a valid Tunnel Encapsulation Limit destination header with the Tun Encap Lim field set to a value zero:

Type 1 - unreachable node

Code 3 - address unreachable

For tunnel ICMP error message "packet too big":

Type 2 - packet too big

Code 0

MTU The MTU field from the tunnel ICMP message minus the length of the tunnel headers.

According to the general rules described in 7.1, an ICMP "packet too big" message is sent to the source of the original packet only if the original packet size is larger than the minimum link MTU size required for IPv6 [IPv6-Spec].

8.3 ICMP Messages for IPv4 Original Packets

The tunnel entry-point node builds the ICMP and IPv4 header of the ICMP message that is sent to the source of the original packet as follows:

IPv4 Fields:

Source Address

A valid unicast IPv4 address of the outgoing interface.

Destination Address

Copied from the Source Address field of the Original IPv4 header.

ICMP Fields:

For any of the following tunnel ICMP error messages:

"hop limit exceeded"

"unreachable node"

"parameter problem" - pointing to a valid Tunnel Encapsulation Limit destination header with the Tun Encap Lim field set to a value zero:

Type 3 - destination unreachable

Code 1 - host unreachable

For a tunnel ICMP error message "packet too big":

Type 3 - destination unreachable

Code 4 - packet too big

MTU The MTU field from the tunnel ICMP message minus the length of the tunnel headers.

According to the general rules described in section 7.2, an ICMP "packet too big" message is sent to the original IPv4 packet source node if the the original IPv4 header has the DF - don't fragment - bit flag SET.

8.4 ICMP Messages for Nested Tunnel Packets

In case of an error uncovered with a nested tunnel packet, the inner tunnel entry-point, which receives the ICMP error message from the inner tunnel reporting node, relays the ICMP message to the outer tunnel entry-point following the mechanisms described in sections 8., 8.1, 8.2, and 8.3. Further, the outer tunnel entry-point relays the ICMP message to the source of the original packet, following the same mechanisms.

9. Security Considerations

An IPv6 tunnel can be secured by securing the IPv6 path between the tunnel entry-point and exit-point node. The security architecture, mechanisms, and services are described in [RFC2401], [RFC2402], and [RFC2406]. A secure IPv6 tunnel may act as a gateway-to-gateway secure path as described in [RFC2401].

For a secure IPv6 tunnel, in addition to the mechanisms described earlier in this document, the entry-point node of the tunnel performs security algorithms on the packet and prepends as part of the tunnel headers one or more security headers in conformance with [IPv6-Spec], [RFC2401], and [RFC2402], or [RFC2406].

The exit-point node of a secure IPv6 tunnel performs security algorithms and processes the tunnel security header[s] as part of the tunnel headers processing described earlier, and in conformance with [RFC2401], and [RFC2402], or [RFC2406]. The exit-point node discards the tunnel security header[s] with the rest of the tunnel headers after tunnel headers processing completion.

The degree of integrity, authentication, and confidentiality and the security processing performed on a tunnel packet at the entry-point and exit-point node of a secure IPv6 tunnel depend on the type of security header - authentication (AH) or encryption (ESP) - and parameters configured in the Security Association for the tunnel. There is no dependency or interaction between the security level and mechanisms applied to the tunnel packets and the security applied to the original packets which are the payloads of the tunnel packets. In case of nested tunnels, each inner tunnel may have its own set of security services, independently from those of the outer tunnels, or of those between the source and destination of the original packet.

10. Acknowledgments

This document is partially derived from several discussions about IPv6 tunneling on the IPng Working Group Mailing List and from feedback from the IPng Working Group to an IPv6 presentation that focused on IPv6 tunneling at the 33rd IETF, in Stockholm, in July 1995.

Additionally, the following documents that focused on tunneling or encapsulation were helpful references: RFC 1933 (R. Gilligan, E. Nordmark), RFC 1241 (R. Woodburn, D. Mills), RFC 1326 (P. Tsuchiya), RFC 1701, RFC 1702 (S. Hanks, D. Farinacci, P. Traina), RFC 1853 (W. Simpson), as well as RFC 2003 (C. Perkins).

Brian Carpenter, Richard Draves, Bob Hinden, Thomas Narten, Erik Nordmark (in alphabetical order) gave valuable reviewing comments and suggestions for the improvement of this document. Scott Bradner, Ross Callon, Dmitry Haskin, Paul Traina, and James Watt (in alphabetical order) shared their view or experience on matters of concern in this document. Judith Grossman provided a sample of her many years of editorial and writing experience as well as a good amount of probing technical questions.

11. References

[IPv6-Spec] Deering, S. and R. Hinden, "Internet Protocol Version 6 (IPv6) Specification", RFC 2460, December 1998.

- [ICMP-Spec] Conta, A. and S. Deering "Internet Control Message Protocol for the Internet Protocol Version 6 (IPv6)", RFC 2463, December 1998.
- [ND-Spec] Narten, T., Nordmark, E., and W. Simpson "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.
- [PMTU-Spec] McCann, J., Deering, S. and J. Mogul, "Path MTU Discovery for IP Version 6 (IPv6)", RFC 1981, August 1996.
- [RFC2401] Atkinson, R., "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [RFC2402] Atkinson, R., "IP Authentication Header", RFC 2402, November 1998.
- [RFC2406] Atkinson, R., "IP Encapsulation Security Payload (ESP)", RFC 2406, November 1998.
- [RFC-1853] Simpson, W., "IP in IP Tunneling", RFC 1853, October 1995.
- [Assign-Nr] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700, October 1994. See also:
<http://www.iana.org/numbers.html>
- [RFC2119] Bradner, S., "Key words for use in RFCs to indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Authors' Addresses

Alex Conta
Lucent Technologies Inc.
300 Baker Ave
Concord, MA 01742-2168
+1-978-287-2842

EMail: aconta@lucent.com

Stephen Deering
Cisco Systems
170 West Tasman Dr
San Jose, CA 95132-1706

Phone: +1-408-527-8213
EMail: deering@cisco.com

Appendix A

A.1 Risk Factors in Nested Encapsulation

Nested encapsulations of a packet become a recursive encapsulation if the packet reenters an outer tunnel before exiting it. The cases which present a high risk of recursive encapsulation are those in which a tunnel entry-point node cannot determine whether a packet that undergoes encapsulation reenters the tunnel before exiting it. Routing loops that cause tunnel packets to reenter a tunnel before exiting it are certainly the major cause of the problem. But since routing loops exist, and happen, it is important to understand and describe, the cases in which the risk for recursive encapsulation is higher.

There are two significant elements that determine the risk factor of routing loop recursive encapsulation:

- (a) the type of tunnel,
- (b) the type of route to the tunnel exit-point, which determines the packet forwarding through the tunnel, that is, over the tunnel virtual-link.

A.1.1 Risk Factor in Nested Encapsulation - type of tunnel.

The type of tunnels which were identified as a high risk factor for recursive encapsulation in routing loops are:

"inner tunnels with identical exit-points".

Since the source and destination of an original packet is the main information used to decide whether to forward a packet through a tunnel or not, a recursive encapsulation can be avoided in case of a single tunnel (non-inner), by checking that the packet to be encapsulated is not originated on the entry-point node. This mechanism is suggested in [RFC-1853].

However, this type of protection does not seem to work well in case of inner tunnels with different entry-points, and identical exit-points.

Inner tunnels with different entry-points and identical exit-points introduce ambiguity in deciding whether to encapsulate a packet, when a packet encapsulated in an inner tunnel reaches the entry-point node of an outer tunnel by means of a routing loop. Because the source of the tunnel packet is the inner tunnel entry-point node which is different than the entry-point node of the outer tunnel, the source

address checking (mentioned above) fails to detect an invalid encapsulation, and as a consequence the tunnel packet gets encapsulated at the outer tunnel each time it reaches it through the routing loop.

A.1.2 Risk Factor in Nested Encapsulation - type of route.

The type of route to a tunnel exit-point node has been also identified as a high risk factor of recursive encapsulation in routing loops.

One type of route to a tunnel exit-point node is a route to a specified destination node, that is, the destination is a valid specified IPv6 address (route to node). Such a route can be selected based on the longest match of an original packet destination address with the destination address stored in the tunnel entry-point node routing table entry for that route. The packet forwarded on such a route is first encapsulated and then forwarded towards the tunnel exit-point node.

Another type of route to a tunnel exit-point node is a route to a specified prefix-net, that is, the destination is a valid specified IPv6 prefix (route to net). Such a route can be selected based on the longest path match of an original packet destination address with the prefix destination stored in the tunnel entry-point node routing table entry for that route. The packet forwarded on such a route is first encapsulated and then forwarded towards the tunnel exit-point node.

And finally another type of route to a tunnel exit-point is a default route, or a route to an unspecified destination. This route is selected when no other match for the destination of the original packet has been found in the routing table. A tunnel that is the first hop of a default route is a "default tunnel".

If the route to a tunnel exit-point is a route to node, the risk factor for recursive encapsulation is minimum.

If the route to a tunnel exit-point is a route to net, the risk factor for recursive encapsulation is medium. There is a range of destination addresses that will match the prefix the route is associated with. If one or more inner tunnels with different tunnel entry-points have exit-point node addresses that match the route to net of an outer tunnel exit-point, then a recursive encapsulation may occur if a tunnel packet gets diverted from inside such an inner tunnel to the entry-point of the outer tunnel that has a route to its exit-point that matches the exit-point of an inner tunnel.

If the route to a tunnel exit-point is a default route, the risk factor for recursive encapsulation is maximum. Packets are forwarded through a default tunnel for lack of a better route. In many situations, forwarding through a default tunnel can happen for a wide range of destination addresses which at the maximum extent is the entire Internet minus the node's link. As consequence, it is likely that in a routing loop case, if a tunnel packet gets diverted from an inner tunnel to an outer tunnel entry-point in which the tunnel is a default tunnel, the packet will be once more encapsulated, because the default routing mechanism will not be able to discern differently, based on the destination.

Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

NEMO Working Group
INTERNET DRAFT
Category: Standards Track
Expires March 2004

Vijay Devarapalli
Nokia
Ryuji Wakikawa
Keio University
Alexandru Petrescu
Motorola
Pascal Thubert
Cisco Systems
September 2003

Nemo Basic Support Protocol
draft-ietf-nemo-basic-support-01.txt

Status of This Memo

This document is an Internet-Draft and is subject to all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at:

<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at:

<http://www.ietf.org/shadow.html>.

Abstract

This document describes the Nemo Basic Support protocol to support network mobility as the mobile network attaches to different points in the Internet. The protocol is based on extensions to Mobile IPv6 and allows for session continuity for every node in the mobile network as the network moves. It also allows every node in the mobile network to be reachable while moving around. The Mobile Router, which connects the network to the Internet, runs the NEMO Basic Support protocol with its Home Agent. The protocol is designed in such a way that network mobility is transparent to the nodes inside the mobile network.

Contents

Status of This Memo	1
Abstract	1
1. Introduction	4
2. Terminology	5
3. Overview of the Nemo Protocol	6
4. Message Formats	9
4.1. Binding Update	9
4.2. Binding Acknowledgement	9
4.3. Mobile Network Prefix Option	10
4.4. Mobile Network Prefix Length Option	11
5. Mobile Router Operation	13
5.1. Data Structures	13
5.2. Sending Binding Updates	13
5.3. Receiving Binding Acknowledgements	14
5.4. Error Processing	15
5.5. Establishment of Bi-directional Tunnel	16
5.6. Neighbour Discovery for Mobile Router	17
5.7. Multicast Groups for Mobile Router	17
6. Home Agent Operation	18
6.1. Data Structures	18
6.1.1. Binding Cache	18
6.1.2. Prefix Table	18
6.2. Mobile Network Prefix Registration	19
6.3. Advertising Mobile Network Reachability	20
6.4. Establishment of Bi-directional Tunnel	21
6.5. Forwarding Packets	21
6.6. Sending Binding Acknowledgements	22
6.7. Mobile Network Prefix De-Registration	22
7. Support for Dynamic Routing Protocols	23
8. Security Considerations	25
9. IANA Considerations	25
10. Contributors	25

11. Acknowledgements	26
A. Examples of Operation	28
B. Changes from Previous Version	31
Addresses	32

1. Introduction

This document describes protocol extensions to Mobile IPv6 (MIPv6) [1] to enable support for network mobility. The extensions provide backward compatibility with Mobile IPv6, and in particular, a Nemo compliant Home Agent can operate as a MIPv6 Home Agent as well.

The Nemo Basic Support works in such a way that session continuity is ensured for all the nodes in the mobile network even as the Mobile Router changes its point of attachment to the Internet. It also provides connectivity and reachability for all nodes in the mobile network as the network moves. The solution supports both Local Fixed Nodes [8] and Mobile Nodes in the Mobile Network.

Within the context of this document, the definition of a Mobile Router extends that of a Mobile IPv6 [1] Mobile Node, by adding the capability of routing between its point of attachment (Care-of Address) and a subnet which moves with the Mobile Router.

The solution described in this document requires setting up a bi-directional tunnel between the Mobile Router and its Home Agent. This tunnel is set up when the Mobile Router sends a successful Binding Update to its Home Agent, informing the Home Agent of its current point of attachment.

All traffic between the nodes in the Mobile Network and Correspondent Nodes passes through the Home Agent. This document does not describe how to route optimize this traffic.

The terminology document [8] describes Nested Mobility as a scenario where a Mobile Router allows another Mobile Router to attach to its mobile network. There could be arbitrary levels of nested mobility. The operation of each Mobile Router remains the same whether the Mobile Router attaches to another Mobile Router or to a fixed Access Router on the Internet. The solution described here does not place any restriction on the number of levels for nested mobility. But it should be noted that this might introduce significant overhead on the data packets as each level of nestedness introduces another IPv6 header encapsulation.

2. Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [2].

There is a separate NEMO terminology document [8], which defines the terms related to Network Mobility used in the document.

Prefix Table

It is a list of a Mobile Network Prefixes indexed by the Home Address of a Mobile Router. The prefix table is managed by the Home Agent and is used by the Home Agent to determine which Mobile Network Prefixes are owned a particular Mobile Router. This is an optional data structure.

3. Overview of the Nemo Protocol

A Mobile Network is a network segment or subnet which can move and attach to arbitrary points in the Internet. A Mobile Network does not allow any transit traffic and can only be accessed via specific gateways called Mobile Routers that manage its movement. A Mobile Router does not distribute the Mobile Network routes to the infrastructure at its point of attachment (i.e. in the visited network). Instead, it maintains a bidirectional tunnel to a Home Agent that advertises an aggregation of Mobile Networks to the infrastructure. The Mobile Router is also the default gateway for the Mobile Network.

A Mobile Network can also consist of multiple and nested subnets. A router with no support for mobility may be permanently attached to a Mobile Network for local distribution. Also, Mobile Routers may be attached to Mobile Networks owned by different Mobile Routers and form a graph. In particular, with Basic Nemo Support, each Mobile Router is attached to another Mobile Network by a single interface, and if loops are avoided, the graph is a tree.

A Mobile Router has an unique Home Address through which it is always reachable. The Home Address is configured from a prefix that is aggregated and advertised by its Home Agent. The prefix could either be the prefix advertised on the home link or the prefix delegated to the Mobile Router. The Mobile Router can have more than one Home Address if there are multiple prefixes in the home link. The Mobile Router also advertises one or more prefixes in the mobile network attached to it. The actual mechanism for allocating these Mobile Network Prefixes is outside the scope of this specification.

When the Mobile Router moves away from the home link and attaches to a new access router, it acquires a Care-of Address from the visited link. The Mobile Router at any time can appear and behave as a Mobile Host or a Mobile Router. If the Mobile Router wants connectivity, reachability and session continuity for nodes in the Mobile Network, it acts as a Mobile Router. In either case, as soon as the Mobile Router acquires a Care-of Address, it immediately sends a Binding Update to its Home Agent as described in [1]. When the Home Agent receives this Binding Update it creates a binding cache entry binding the Mobile Router's Home Address to its Care-of address at the current point of attachment.

If the Mobile Router wishes to act as a Mobile Router and provide connectivity to nodes in the Mobile Network, it indicates this to the Home Agent by setting a flag 'R' in the Binding Update. It MAY also include information about the Mobile Network Prefix in the Binding Update using one of the modes described in section 5.2, so that the Home Agent can forward packets meant for nodes in the mobile

network to the Mobile Router. Two new Mobility Header Options are described in this document to carry prefix information. These new options are described in section 4.3 and section 4.4. If the Mobile Network has more than one IPv6 prefix and wants the Home Agent to setup forwarding for all these prefixes, it includes multiple prefix information options in a single Binding Update. The Home Agent sets up forwarding for each of these prefixes to the Mobile Router's Care-of Address. In some scenarios the Home Agent already knows which prefixes are owned by a Mobile Router. In these scenarios, the Mobile Router does not include any prefix information in the Binding Update. The Home Agent sets up forwarding for all prefixes owned by the Mobile Router, when it receives a Binding Update from the mobile router with the router flag 'R' set.

The Home Agent acknowledges the Binding Update by sending a Binding Acknowledgement to the Mobile Router. A positive acknowledgement means that the HA has set up forwarding for the Mobile Network. Once the binding process completes, a bi-directional tunnel is established between the Home Agent and the Mobile Router. The tunnel end points are Mobile Router's Care-of Address and the Home Agent's address. If a packet with a source address belonging to the Mobile Network Prefix is received from the Mobile Network, the Mobile Router reverse-tunnels the packet to the Home Agent through this tunnel. This reverse-tunneling is done by using IP-in-IP encapsulation [3]. The Home Agent decapsulates this packet and forwards it to the Correspondent Node. The Mobile Router is however free to use route optimization as described in [1] for packet originated by the Mobile Router itself.

When a data packet is sent by a Correspondent Node to a node in the Mobile Network, it gets routed to the Home Agent which currently has the binding for the Mobile Router. It is expected that the Mobile Router's network prefix would be aggregated at the Home Agent, which advertises the resulting aggregation. Alternatively, the Home Agent may receive the data packets meant for the Mobile Network by advertising routes to the Mobile Network prefix. The actual mechanism by which these routes are advertised is outside the scope of this document. When the Home Agent receives a data packet meant for a node in the mobile network, it tunnels the packet to Mobile Router's current Care-of address. The Mobile Router decapsulates the packet and forwards it onto the interface where the Mobile Network is connected. The Mobile Router before decapsulating the tunneled packet, has to check if the Source address on the outer IPv6 header is the Home Agent's address. It also has to make sure the destination address on the inner IPv6 header belongs to one of its Mobile Network Prefixes before forwarding the packet to the Mobile Network.

The Mobile Network could consist of nodes which are Local Fixed Nodes, Local Mobile Nodes and Visiting Mobile Nodes [8]. The protocol described here ensures complete transparency of network mobility to the Local Fixed Nodes. Visiting Mobile Nodes are those nodes which are Mobile Nodes as described in Mobile IPv6. Visiting Mobile Nodes treat the Mobile Network as just a normal IPv6 access network and run the Mobile IPv6 protocol.

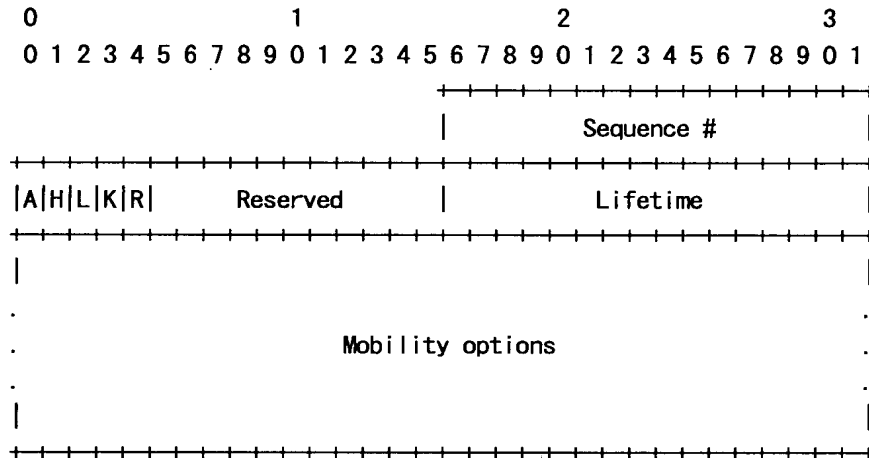
It is also possible for the Mobile Router and the Home Agent to run a routing protocol through the bi-directional tunnel. In that case, the Mobile Router need not include prefix information in the Binding Update. Instead the Home Agent uses the routing protocol updates to setup forwarding for the Mobile Network. When running the routing protocol it is required that the bi-directional tunnel be treated as a tunnel interface. The tunnel interface is included as the list of interfaces on which routing protocol is active. The Mobile Router should be configured not to run the routing protocol on its egress interface when it is away from the home link.

Finally, the Home Agent may be configured with static routes to the Mobile Network Prefix via the Mobile Router's Home Address. In that case, the routes are set independently of the binding flows and the returning Home of a Mobile Router. The benefit is that such movement does not induce any additional signalling in the form of routing updates in the Home Network. The drawback of that model is that the routes are present even if the related Mobile Routers that are not reachable (at Home or bound) at a given point of time.

4. Message Formats

4.1. Binding Update

A new flag 'R' is included in the Binding Update to indicate to the Home Agent if the Binding Update is coming from a Mobile Router and not from a mobile node. The rest of the Binding Update format remains the same as defined in [1].



Mobile Router Flag (R)

The Mobile Router Flag is set to indicate to the Home Agent that the Binding Update is from a Mobile Router. If the flag is set to 0, the Home Agent assumes that the Mobile Router is just behaving as a Mobile Node, and MUST NOT forward packets destined for the mobile network to the Mobile Router.

Mobility Options

Variable length field which can include zero or more mobility options. This document defines two new mobility options in addition to what is defined in [1]. The receiver MUST skip and ignore any options which it does not understand.

4.2. Binding Acknowledgement

There is no change in the Binding Acknowledgement format from what is used in Mobile IPv6 [1]. However, this document introduces the following new status values for the binding acknowledgement.

Status

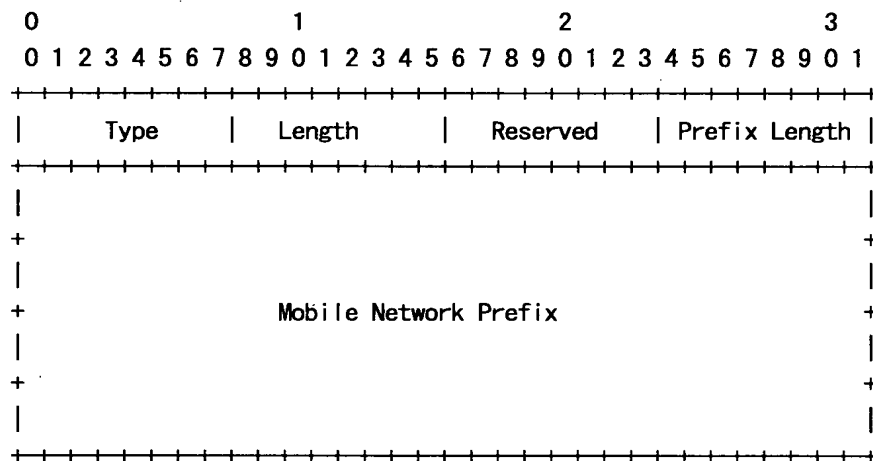
- 2 Mobile Router Binding Update accepted
- 140 Mobile Router Operation not permitted
- 141 Invalid Prefix
- 142 Not Authorized for Prefix
- 143 Forwarding Setup failed

Status values less than 128 indicate that the Binding Update was processed successfully by the receiving nodes. Values greater than 128 indicate that the Binding Update was rejected by the receiving node.

4.3. Mobile Network Prefix Option

The Mobile Network Prefix Option is included in the Binding Update to indicate to the Home Agent the prefix information for the mobile network. There could be multiple Mobile Network Prefix Options if the Mobile Router has more than one IPv6 prefix in the Mobile Network and wants the Home Agent to forward packets for each of these prefixes to the Mobile Router's current location.

The Mobile Network Prefix Option has an alignment requirement of $8n+4$. Its format is as follows.



Type

TBA

Length

8 bit unsigned integer indicating the length in octets of the option excluding the type and length fields. Set to 18.

Reserved

This field is unused for now. The value **MUST** be initialized to zero by the sender, and **MUST** be ignored by the receiver.

Prefix Length

8 bit unsigned integer indicating the prefix length of the IPv6 prefix contained in the option.

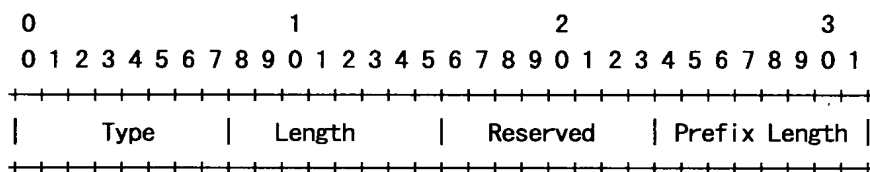
Mobile Network Prefix

A 16 byte field contains the Mobile Network Prefix.

4.4. Mobile Network Prefix Length Option

The Mobile Network Prefix Length Option can be used by the Mobile Router if the Mobile Network Prefix can be deduced from the Home Address of the Mobile Router. If there is only one Mobile Network Prefix owned by the Mobile Router, using this option helps in saving 16 bytes in the Binding Update by not including the prefix information.

There can only be one instance of this option in a Binding Update. The Mobile Network Prefix Option cannot be present in the Binding Update if this option is present.



Type

TBA

Length

8 bit unsigned integer indicating the length in octets of the option excluding the type and length field. Set to 2.

Reserved

This field is unused for now. The value **MUST** be initialized to zero by the sender, and **MUST** be ignored by the receiver.

Prefix Length

8 bit unsigned integer indicating the prefix length of the IPv6 prefix from which the Home Address included in the Binding Update was configured.

5. Mobile Router Operation

Mobile Router operation is derived largely from the combined behaviors of a Host, of a Router [6], and of a Mobile Node [1] (also please see definition of a Mobile Host in [8]).

A Mobile Node can act in two different ways: (1) as a Mobile Host (in which case the Mobile IPv6 Home Agent doesn't maintain any prefix information related to the Mobile Host's Home Address, but does maintain a binding cache entry related to the Mobile Host's Home Address) and (2) as a Mobile Router (in which case, in addition to maintaining the binding cache entry corresponding to the Mobile Router Home Address, the Mobile IPv6 Home Agent also maintains forwarding information related to prefixes assigned to the mobile network). The distinction between the two modes is represented by the value of the Mobile Router flag 'R'.

5.1. Data Structures

Like a Mobile Host, a Mobile Router also maintains a Binding Update List, described in section 11.1 of Mobile IPv6 specification[1]. The Binding Update list is a conceptual data structure which records information that is sent in the Binding Updates. There is one entry per each destination that the Mobile Router is currently sending Binding Updates to.

This document introduces a new Prefix Information field in the Binding Update list structure. This field is used to store any prefix information that the Mobile Router includes in the Binding Update. If the Mobile Router sets the Mobile Router flag 'R' in the Binding Update, but does not include any prefix information in it (implicit mode), this field is set to null.

Similar to a Mobile Host, a Mobile Router stores the information regarding status of flags of the Binding Update, in the corresponding Binding Update List entry. This document introduces a new mobile router flag 'R' for this entry. The status of this flag is stored in the Binding Update list whenever a Binding Update is sent.

A Mobile Router also maintains a Home Agent list populated according to the same procedure as a Mobile Host.

5.2. Sending Binding Updates

A Mobile Router sends Binding Updates to its Home Agent according to the same procedures that a Mobile Host uses. The Mobile Router uses one of the following modes to instruct the Home Agent to determine

the prefixes owned by the Mobile Router. In all three modes, the Mobile Router sets the Mobile Router flag 'R'.

Implicit:

In this mode, the Mobile Router does not include either a Mobile Network Prefix Option or a Mobile Network Prefix Length Option in the Binding Update (but it does include the Home Address Option in the Destination Options header, as all Mobile Hosts do). The Home Agent can use any mechanism (not defined in this document) to determine the Mobile Network Prefix(es) owned by the Mobile Router and setup forwarding for the Mobile Network. One example would be manual configuration at the Home Agent mapping the Mobile Router's Home Address to the information required for setting up forwarding for the Mobile Network.

Explicit Network:

In this mode, the Mobile Router includes one or more Mobile Network Prefix Options in the Binding Update. These options contain information about the Mobile Network Prefix(es) configured on the mobile network.

Explicit Prefix Length:

In this mode, the Mobile Router instructs the Home Agent to derive the Mobile Network Prefix by using: (1) the Home Address in the Home Address Option carried in the Destination Options header of the same packet that carries the Mobility Header containing this Binding Update and (2) the prefix length carried in the Mobile Network Prefix Length Option. In this case, Mobile Router includes one and only one Mobile Network Prefix Length Option. It MUST NOT include a Mobile Network Prefix Option if this method is used.

If the Mobile Router flag is set, Home Registration flag 'H' MUST be set.

5.3. Receiving Binding Acknowledgements

The Mobile Router receives Binding Acknowledgements from the Home Agent, corresponding to the Binding Updates it sent. If the Binding Acknowledgement status is set to '2' (Mobile Router Binding Update accepted), the Mobile Router assumes that the Home Agent has successfully processed the Binding Update and has set up forwarding for the Mobile Network. The Mobile Router can then start using the

bi-directional tunnel for reverse tunneling traffic from the mobile network.

5.4. Error Processing

If the Binding Acknowledgement status is set to a value between 128 and 140, the Mobile Router takes necessary actions as described in the Mobile IPv6 specification [1].

If the Binding Acknowledgement status is set to '0' (Binding Update accepted), the Mobile Router concludes that the Home Agent which processed the Binding Update is a MIPv6 Home Agent that has not implemented support for Mobile Routers. It should send a similar Binding Update to another Home Agent on the link. If no Home Agent replies positively then the Mobile Router **MUST** refrain from sending any Binding Update with the Mobile Router flag set to any home agent on the home link and log the information.

If the Mobile Router sent a Binding Update to the Home Agent in implicit mode (i.e. the prefix field in the Binding Update list entry is null) then the Mobile Router interprets only the error status '140' (Mobile Router Operation not permitted) and '143' (Forwarding Setup failed) For this Binding Update, the Mobile Router **MUST** discard Binding Acknowledgements with codes '141' and '142'.

For the same Binding Update, if the status is '140', then the Mobile Router should send a similar Binding Update (implicit mode) to another Home Agent on the same home link. If no Home Agent replies positively then the Mobile Router **MUST** refrain from sending any Binding Update with the Mobile Router flag set to any Home Agent on the home link, and log the information.

For the same Binding Update, if the status is '143', then the Mobile Router should send a similar Binding Update (implicit mode) to another Home Agent on the same home link. If no Home Agent replies positively then Mobile Router **SHOULD** refrain from sending this Binding Update to any Home Agent on the home link, and **MAY** send Binding Updates in another mode (e.g. explicitly include a prefix) to a Home Agent on the same home link.

If the Mobile Router sent a Binding Update to Home Agent in any other mode than implicit mode (i.e. the prefix field in the Binding Update list entry is not null) then the Mobile Router interprets only the error status '141' (Invalid Prefix) and '142' (Not Authorized for Prefix). For this Binding Update, the Mobile Router **MUST** discard Binding Acknowledgements with codes '140' and '143'.

For the same Binding Update, if the status is set to '141', then the Mobile Router should send a similar Binding Update (same explicit prefix(es) or prefix lengths) to another Home Agent on the same home link. If no Home Agent replies positively then Mobile Router SHOULD refrain from sending this Binding Updates to any Home Agent on the home link. At this point, Mobile Router MAY try to obtain and own a prefix by the same means that it initially got attributed the Invalid Prefix in question. Alternatively, Mobile Router MAY send Binding Updates in another mode (e.g. implicit mode) to a Home Agent on the same home link.

For the same Binding Update, if the status is set to '142', then the Mobile Router should send a similar Binding Update (same explicit prefix(es) or prefix lens) to another Home Agent on the same home link. If no Home Agent replies positively then Mobile Router SHOULD refrain from sending this Binding Updates to any Home Agent on the home link. Additionally, the Home Agent MUST stop advertising the respective prefix(es) in the mobile network with associated Router Advertisements, and modify its own forwarding information accordingly. Following this, the Mobile Router MAY send Binding Updates in another mode (e.g. implicit) to a Home Agent on the same home link.

If at the end of this Error Processing procedure the Mobile Router has tried every available modes of sending Binding Updates and still has not received a positive Binding Acknowledgement (status value between 0 and 127) for this Home Address from any Home Agent on its home link, then the Mobile Router MUST stop sending Binding Updates with the Mobile Router flag set for this Home Address and log the information.

In all the above cases, the Mobile Router MUST conclude that the Home Agent did not create a binding cache entry for the Mobile Router's Home Address.

5.5. Establishment of Bi-directional Tunnel

When a successful Binding Acknowledgement with status set to '2' (Mobile Router Binding Update accepted) is received, the Mobile Router set up its endpoint of the bi-directional tunnel.

The bi-directional tunnel between Mobile Router and Home Agent allows packets to flow in both directions between these entities, while the Mobile Router is connected to a Visited Link. The bi-directional tunnel involves two virtual links [3]: one virtual link has the address of the tunnel entry point as the Care-of Address of the Mobile Router and the tunnel exit point as the address of the Home Agent; the other virtual link has as tunnel entry point the

Home Agent address and as tunnel exit point the Care-of Address of the Mobile Router. Both addresses are unicast addresses. All IPv6 traffic to and from the Mobile Network is sent through this bi-directional tunnel.

A Mobile Router MAY limit the number of mobile routers that attach to its mobile network (the number of levels in the nested aggregation) by means of setting the Tunnel Encapsulation Limit field of the Tunnel Encapsulation option.

A Mobile Router uses the Tunnel Hop Limit that is normally assigned to routers (not to hosts). See [3].

5.6. Neighbour Discovery for Mobile Router

A Mobile Router MAY be configured to send Router Advertisements and reply to Router Solicitations on the interface attached to the home link. The value of the Router Lifetime field MUST be set to zero to prevent other nodes from configuring the Mobile Router as the default router.

A Mobile Router SHOULD NOT send unsolicited Router Advertisements and SHOULD NOT reply to Router Solicitations on any egress interface when that interface is attached to a visited link. However, the Mobile Router SHOULD reply with Neighbor Advertisements to Neighbor Solicitations received on the egress interface, for topologically correct addresses.

A Mobile Router MUST NOT ignore Router Advertisements received on the egress interface. The received Router Advertisements MAY be used for address configuration, default router selection or movement detection.

5.7. Multicast Groups for Mobile Router

When at home, the Mobile Router joins the multicast group All Routers Address with scopes '1' interface-local (on the home-advertising interface) and '2' link-local on any of its egress interfaces. When in a visited network, the Mobile Router MUST NOT join the above multicast groups on the corresponding interface.

6. Home Agent Operation

In order for a Mobile Router to operate correctly, the Home Agent **MUST** satisfy all the requirements listed in section 8.4 of [1]. The Home Agent **MUST** implement all the modes described in Section 5.2.

6.1. Data Structures

6.1.1. Binding Cache

The Home Agent maintains Binding Cache Entries for each Mobile Router that is currently registered with the Home Agent. The Binding Cache is a conceptual data structure described in detail in [1].

The Home Agent might need to store the Mobile Network Prefixes associated with a Mobile Router in the corresponding Binding Cache Entry. This is required if the Binding Update (that created the Binding Cache Entry) contained explicit prefix information. This information can be used later to cleanup routes installed in explicit mode, when the Binding Cache Entry is removed, and to maintain the routing table, for instance should the routes be manually removed.

The Home Agent also stores the status of the Mobile Router Flag 'R' in the Binding Cache entry.

6.1.2. Prefix Table

In some deployment scenarios it may be necessary for the Home Agent to prevent a misbehaving Mobile Router from claiming mobile network prefixes belonging to another Mobile Router. The Home Agent can prevent such attacks if it maintains a Prefix Table and verifies the Prefix Information provided by the Mobile Router against the entries in the Prefix Table. However, this verification is done only if the Binding Update contained explicit prefix information in the form of either the Mobile Network Prefix Option or the Mobile Network Prefix Length Option.

Each entry in the Prefix Table conceptually contains the following fields:

- The Home Address of the Mobile Router. This field is used as the key for searching the pre-configured prefix table.
- The Mobile Network Prefix of the Mobile Router associated with the Home Address.

6.2. Mobile Network Prefix Registration

The Home Agent processes the Binding Update as described in section 10.3.1 of the Mobile IPv6 specification [1]. This section describes the processing of the Binding Update if the Mobile Router (R) flag is set. The Home Agent performs the following check in addition.

- The Home Registration (H) flag MUST be set. If not, the Home Agent MUST reject the Binding Update and send a Binding Acknowledgement with status set to 140. Note: The basic support does not allow sending Binding Update for a Mobile Network Prefix to correspondent nodes (for route optimization).
- If the Mobile Network Prefix Length option is present in the Binding Update, then there MUST be only one instance of this option in the Binding Update. Also the Mobile Network Prefix Option MUST NOT be present in the same Binding Update. Otherwise, the Home Agent MUST discard the Binding Update and send an ICMP Parameter Problem, Code 0, message to the Mobile Router.

If the Home Agent does not reject the Binding Update as described above, then it retrieves the Mobile Network Prefix information as described below.

- If a Mobile Network Prefix Length Option is present in the Binding Update, the Home Address in the Home Address destination option MUST be an Home Address. In that case, the Mobile Network Prefix is obtained from that Home Address and the prefix length in the Mobile Network Prefix Length Option.

If the Home Agent verifies the prefix information with the Prefix Table and the check fails, the Home Agent MUST discard the Binding Update and send a Binding Acknowledgement with status set to 142 (Not Authorized for Prefix).

- If a Mobile Network Prefix Option is present in the Binding Update, the prefix information for the mobile network prefix is retrieved from the Mobile Network Prefix field and the Prefix Length field of the option. If the Binding Update contains more than one option, the Home Agent MUST set up forwarding for all of the Mobile Network Prefixes. If the Home Agent fails to setup forwarding to all the prefixes listed in the Binding Update, then it MUST NOT forward traffic to any of the prefixes, reject the Binding Update and send a Binding Acknowledgement with status set to 141 (Invalid Prefix).

If the Home Agent verifies the prefix information with the Prefix Table and the check fails, the Home Agent MUST discard the

Binding Update and send a Binding Acknowledgement with status set to 142 (Not Authorized for Prefix).

- If there are no options in the Binding Update, the Home Agent uses manual pre-configured information to determine the prefixes assigned to the Mobile Router and for setting up forwarding for the Mobile Network. If there is no information that the Home Agent can use, it MUST reject the Binding Update and send a Binding Acknowledgement with status set to 143 (Forwarding Setup failed).

If the Lifetime specified in the Binding Update is zero or the specified Care-of address matches the Home Address in the Binding Update, then this is a request to delete the cached binding for the home address and specified Mobile Network Prefixes. The Binding Update is processed according to the procedure described in section 6.7.

If all checks are passed, the Home Agent creates a binding cache entry for Mobile Router's Home Address, or updates the binding cache entry if it already exists. Otherwise, the Home Agent MUST NOT register the binding of the Mobile Router's Home Address.

The Home Agent defends the Mobile Router's Home Address through Proxy Neighbor Discovery by multicasting onto the home link a Neighbor Advertisement message on behalf of the mobile router. All fields in each such Neighbor Advertisement message SHOULD be set in the same way they would be set by the mobile router itself if sending this Neighbor Advertisement while at home, as described in [7], with the exception that the Router (R) bit in the Advertisement MUST be set if the Mobile Router (R) flag has been set in the Binding Update.

The Home Agent also creates a bi-directional tunnel to the mobile router for the requested Mobile Network Prefix, or update an existing bi-directional tunnel as described in section 6.4.

6.3. Advertising Mobile Network Reachability

In order to be able to receive packets meant for the mobile network, the Home Agent advertises reachability to the mobile network. If the Home Link is configured with a prefix that is an aggregation and if the Mobile Network Prefix is aggregated under that prefix, then the routing updates advertising reachability to the mobile network are sent only on the Home Link. If the Home Agent is the only default router on the Home Link, routes to the Mobile Network Prefix get aggregated naturally under the Home Agent and the Home Agent does not have to do anything special.

If the Home Agent receives routing updates through a dynamic routing protocol from the Mobile Router, those routes are propagated by the routing protocol running on the Home Agent on the relevant interfaces.

6.4. Establishment of Bi-directional Tunnel

The establishment and operation of the bi-directional tunnel is implementation specific. However, all implementations MUST be capable of the following operations.

- The Home Agent can tunnel packets meant for the mobile network prefix to the Mobile Router's current location, the Care-of Address of the Mobile Router.
- The Home Agent can accept packets tunneled by the Mobile Router with source address of the outer IPv6 header set to the Care-of Address of the Mobile Router.

6.5. Forwarding Packets

When the Home Agent receives a data packet destined for the mobile network, it forwards the packet to the Mobile Router through the bi-directional tunnel. The Home Agent either uses only the routing table, only the Binding Cache or a combination of routing table and Binding Cache to route packets to the mobile network. This is implementation specific. Two examples are shown below.

1. The Home Agent maintains a route to the Mobile Network Prefix with the next hop set to the Mobile Router's Home Address. When the Home Agent tries to forward the packet to the next hop, it finds a binding cache entry for the home address. Then the Home Agent extracts the Mobile Router's Care-of address and tunnels the packet to the Care-of address.
2. The Home Agent maintains a route to the Mobile Network Prefix with the outgoing interface set to the bi-directional tunnel interface between the Home Agent and the Mobile Router. For this purpose, the Home Agent MUST treat this tunnel as a tunnel interface. When the packets are forwarded through the tunnel interface, they get encapsulated automatically with the source address and destination address in the outer IPv6 header set to the Home Agent's address and the Mobile Router's Care-of address, respectively.

6.6. Sending Binding Acknowledgements

A Home Agent serving a Mobile Router sends Binding Acknowledgements according to the same rules it uses for sending Binding Acknowledgements to Mobile Hosts, with the following enhancements.

The Home Agent sets the status code in the Binding Acknowledgement to '2' (Mobile Router Binding Update accepted) in order to indicate to the Mobile Router that it accepted the Binding Update, set up the tunnel endpoint and the necessary forwarding information.

If the Home Agent is configured not to support mobile routers, it sets the status code in the Binding Acknowledgement to '140' (Mobile Router Operation not permitted).

If one or more prefixes received in the Binding Update are invalid and the Home Agent cannot setup forwarding for the prefixes, the Home Agent sets the status code in the Binding Acknowledgement to '141' (Invalid Prefix) in order to indicate this to the Mobile Router.

If the Mobile Router is not authorized to use this Home Address to forward packets for one or more prefixes that are present in the Binding Update, the Home Agent sets the status code in the Binding Acknowledgement to '142' (Not Authorized for Prefix) in order to indicate this.

The Home Agent sets the status code to 143 (Forwarding Setup failed) if it is unable to determine the information needed to setup forwarding for the Mobile Network. This is used in the Implicit mode where the Mobile Router does not include any prefix information in the Binding Update.

6.7. Mobile Network Prefix De-Registration

The Mobile Router de-registers with the Home Agent by sending a Binding Update with the lifetime set to zero. When the Home Agent successfully processes the de-registration BU, it deletes the Binding Cache Entry for the Mobile Router's Home Address and stops proxying the Home Address. This is described in detail in the Mobile IPv6 specification [1].

In addition, the Home Agent also removes the bi-directional tunnel and stops forwarding packets to the mobile network. The HA should keep all necessary information to clean up whichever routes it installed, whether they come from implicit or explicit source.

7. Support for Dynamic Routing Protocols

In the solution described so far, forwarding to the Mobile Network at the Home Agent is set up when the Home Agent receives a Binding Update from the Mobile Router. An alternative to this is for the Home Agent and the Mobile Router to run an intra-domain routing protocol like RIPng [10] and OSPF [11] through the bi-directional tunnel. The Mobile Router can continue running the same routing protocol that it was running when it was attached to the home link.

This feature is very useful when the Mobile Network is large with multiple subnets containing different IPv6 prefixes. Routing changes in the Mobile Network are propagated to the Home Agent quickly. Routing changes in the home link are also propagated to the Mobile Router very quickly.

When the Mobile Router is attached to the home link, it runs a routing protocol by sending routing updates through its egress interface. When the mobile router moves and attaches to a visited network, it MUST stop sending routing updates on the interface with which it attaches to the visited link. This is very important so that IPv6 prefixes specific to the Mobile Network do not leak into the visited network. The Mobile Router then starts sending routing protocol messages through the bi-directional tunnel towards the Home Agent. Most routing protocols use link local addresses as source addresses for the routing information messages. The Mobile Router is allowed to use link local addresses for the inner IPv6 header of an encapsulated packet. But these messages after decapsulation MUST NOT be forwarded to another link by either the Mobile Router or the Home Agent.

When the Home Agent receives the encapsulated routing protocol message, it processes the inner packets and updates its routing table accordingly. The next hop information in these routing entries is filled with the Mobile Router's link local address with the outgoing interface set to the bi-directional tunnel.

Similarly, the Home Agent also sends routing updates through the bi-directional tunnel to the Mobile Router. The Mobile Router processes these routing protocol messages and updates its routing table. For all routes advertised by the Home Agent, the Mobile Router sets the outgoing interface to the bi-directional tunnel to the Home Agent.

When the Mobile Router and the Home Agent exchange routes through a dynamic routing protocol, the Mobile Router should be careful in including the same mobile network prefixes in the Binding Update to the HA and in the routing protocol updates. The HA depending on its configuration might not add routes based on the prefix information in

the Binding Updates at all, and might use only the routing protocol updates. Moreover, including the same prefix information in both the Binding Update and the routing protocol update is redundant.

The tunneled routing messages **MUST** be authenticated and encrypted by using IPsec ESP [4] in tunnel mode.

8. Security Considerations

All signaling messages between the Mobile Router and the Home Agent MUST be authenticated by IPsec [5]. The use of IPsec to protect Mobile IPv6 signaling messages is described in detail in the HA-MN IPsec specification [2]. The signaling messages described in this document just extend Mobile IPv6 messages and do not require any changes to what is described in the HA-MN IPsec specification.

The Home Agent has to verify that packets received through the bi-directional tunnel belong to the Mobile Network. This check is necessary in order to prevent nodes from using the Home Agent to launch attacks that would have otherwise been prevented by ingress filtering. The source address of the outer IPv6 header MUST be set the Mobile Router's current Care-of address. The source address of the inner IPv6 header MUST belong to the Mobile Network Prefix owned by the Mobile Router.

When the Mobile Router is running a dynamic routing protocol as described in section 7, it injects routing update messages into the Home Link. The Home Agent MUST verify that the Mobile Router is allowed to send routing updates before processing the messages and propagating the routing information.

Please refer to the Mobile IPv6 specification [1] for security considerations when the Mobile Router operates as a Mobile Host.

9. IANA Considerations

This document defines two new Mobility Header Options.

- Mobile Network Prefix Option.
- Mobile Network Prefix Length Option.

These options are described in section 4.3 and section 4.4. The type values for these options need to assigned from the same space used by the mobility options defined in [1]

10. Contributors

We would like to acknowledge Ludovic Bellier, Claude Castelluccia, Thierry Ernst, Miguel Catalina-Gallego, Christophe Janneteau, T.J. Kniveton, Hong-Yon Lach, Jari T. Malinen, Koshiro Mitsuya, Alexis Olivereau, Charles E. Perkins and Keisuke Uehara, for their work on earlier proposals for Network Mobility. This document inherits a lot of ideas from these proposals.

11. Acknowledgements

We thank all members of the NEMO Working Group, and of the preceding MONET BoF for fruitful discussions on the mailing list and at IETF meetings.

Kent Leung, Marco Molteni and Patrick Wetterwald for their work on Network Mobility for IPv4 and IPv6.

Tim Leinmueller for many insightful remarks and implementation aspects.

Normative References

- [1] D. Johnson, C. Perkins and J. Arkko. Mobility Support in IPv6. Internet Draft, IETF. draft-ietf-mobileip-ipv6-24.txt (work in progress). June 2003.
- [2] J. Arkko, V. Devarapalli and F. Dupont. Using IPsec to Protect Mobile IPv6 Signaling between Mobile Nodes and Home Agents. Internet Draft, IETF. draft-ietf-mobileip-mipv6-ha-ipsec-06.txt (work in progress). June 2003.
- [3] A. Conta and S. Deering. Generic Packet Tunneling in IPv6 Specification. RFC 2473, IETF. December 1998.
- [4] S. Kent and R. Atkinson. IP Encapsulating Security Payload (ESP). RFC 2402, IETF. November 1998.
- [5] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401, IETF. November 1998.
- [6] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, IETF. December 1998.
- [7] T. Narten, E. Nordmark and W. Simpson. Neighbour Discovery for IP Version 6 (IPv6). RFC 2461, IETF. December 1998.

Informative References

- [8] T. Ernst and H.-Y. Lach. Network Mobility Support Terminology. Internet Draft, IETF. draft-ietf-nemo-terminology-00.txt (work in progress). May 2003.
- [9] T. Ernst. Network Mobility Support Goals and Requirements. Internet Draft, IETF. draft-ietf-nemo-requirements-01.txt (work in progress). May 2003.
- [10] G. Malkin and R. Minnear. RIPng for IPv6. RFC 2080, IETF. January 1997.
- [11] R. Coltun, D. Ferguson and J. Moy. OSPF for IPv6. RFC 2470, IETF. December 1999.

A. Examples of Operation

This section tries to illustrate the NEMO protocol using a Mobile Router and a Mobile Node belonging to different administrative domains. The Mobile Router's mobile network consists of a Local Fixed Node (LFN) and a Local Fixed Router (LFR) [8]. The LFR has an access link to which other Mobile Nodes or Mobile Routers could attach to.

Figure 1 depicts the scenario where both the Mobile Router and the Mobile Node are at home.

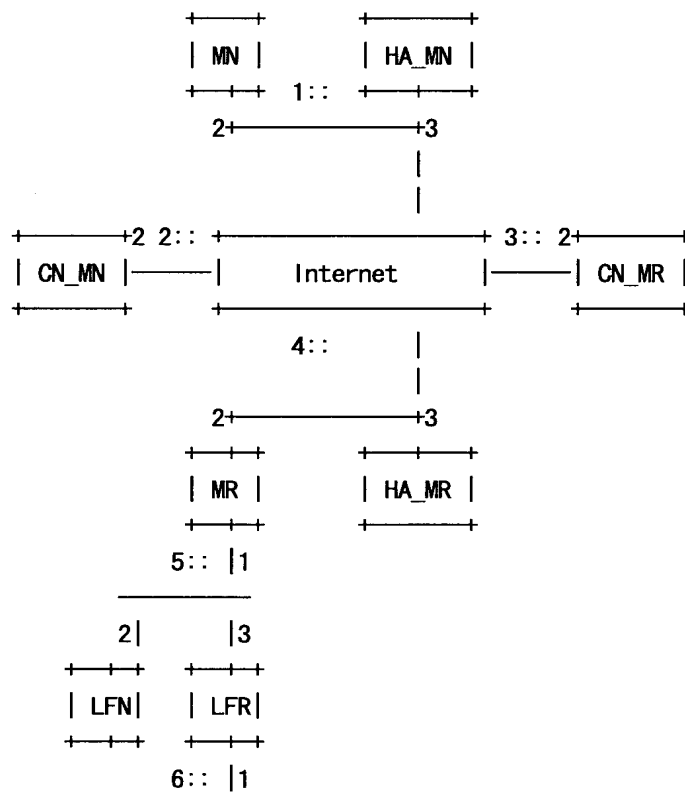


Figure 1: Mobile Router and Mobile Node at home.

The Mobile Router then moves away from the home link and attaches to a visited link. This is shown in Figure 2. The Mobile Router sends a Binding Update to HA_MR when it attaches to a visited link and configures a Care-of Address. HA_MR creates a binding cache entry for the Mobile Router's Home Address and also sets up forwarding for the prefixes on the mobile network.

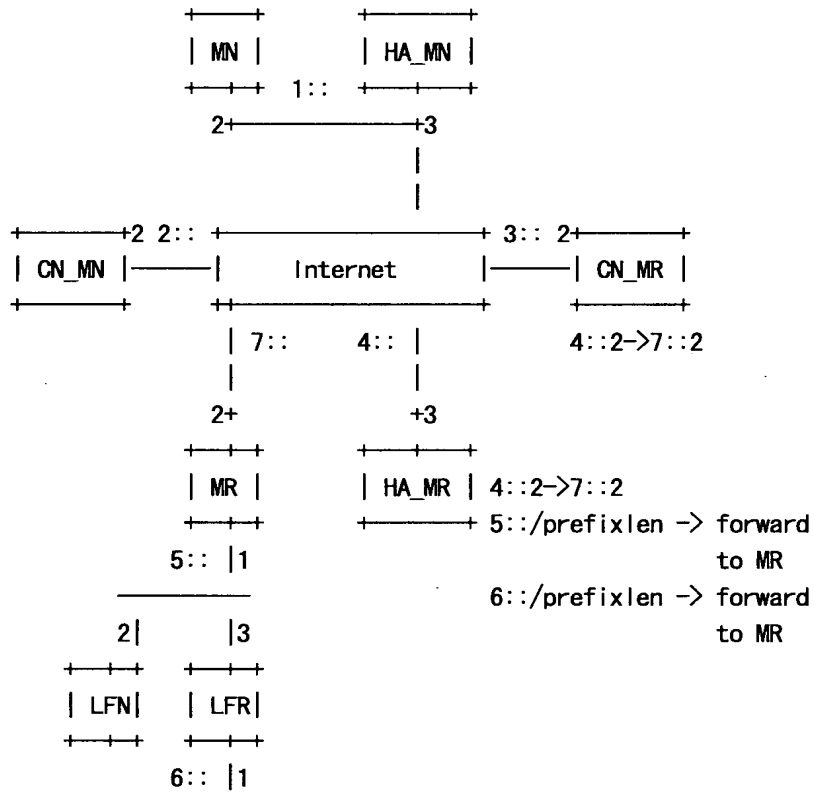


Figure 2: Mobile Router on a Visited Link.

Figure 3 shows the Mobile Node moving away from its home link and attaching to the Mobile Router. The Mobile Node configures a Care-of Address from the prefix advertised on the mobile network and sends a Binding Update to its Home Agent (HA_MN) and its Correspondent Node (CN_MN). Both HA_MN and CN_MN create binding cache entries for the Mobile Node's Home Address.

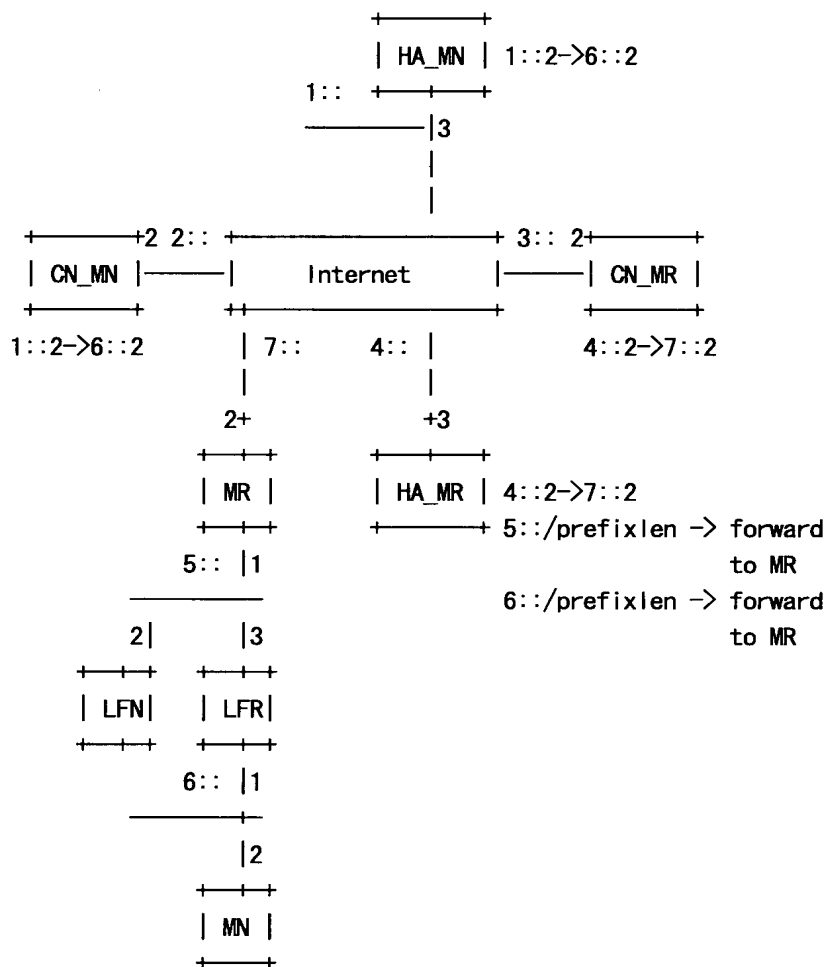


Figure 3: Mobile Node attached to Mobile Router on a Visited Link

B. Changes from Previous Version

The following changes have been made to this document from version 00

- Clarified that Router flag must be set in the Proxy Neighbor Advertisement sent for a Mobile Router by the Home Agent. (Issue 1).
- Clarified that if the Router flag in the Binding Update is set, then the HA should assume that the Mobile Router wants to be treated as a mobile node. (Issue 3).
- Added text to make it clear that the Home Agent must perform ingress filtering on all packets reverse tunneled by the Mobile Router. (Issue 3).
- Extended Home Network concept has been removed from this document. (Issue 5).
- Added text to clarify differences between Implicit mode and running a dynamic routing protocol. (Issue 6).
- Clarified that Prefix Table is used only in explicit mode. In Implicit mode the Prefix Table is not used. (Issue 7 and 12).
- Added text to specify the Home Agent must support all modes. The Mobile Router needs to support only one mode. (Issue 11).
- Added text to support interoperability between a Mobile Router and a legacy MIPv6 Home Agent. (Issue 15).

Authors Addresses

Vijay Devarapalli
Nokia Research Center
313 Fairchild Drive
Mountain View, CA 94043
USA
Email: vijay.devarapalli@nokia.com

Ryuji Wakikawa
Keio University and WIDE
5322 Endo Fujisawa Kanagawa
252-8520 Japan
Email: ryuji@sfc.wide.ad.jp

Alexandru Petrescu
Motorola Labs
Parc les Algorithmes Saint Aubin
Gif-sur-Yvette 91193
France
Email: Alexandru.Petrescu@motorola.com

Pascal Thubert
Cisco Systems Technology Center
Village d'Entreprises Green Side
400, Avenue Rومانille
Biot - Sophia Antipolis 06410
France
Email: pthubert@cisco.com

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.